Cumhur Ozan Çetinaslan

Position Manipulation Techniques for Facial Animation

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
July 2016

Cumhur Ozan Çetinaslan

Position Manipulation Techniques for Facial Animation

Thesis submitted to Faculdade de Ciências da Universidade do Porto

Orientador: Verónica Orvalho
Co-orientador:

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
July 2016

Acknowledgments

First and foremost, I would like to express my gratitude to my advisor Verónica Orvalho for accepting me as a PhD student and giving me her consistent motivation and support during my research in the field of computer graphics. I also would like to thank J.P. Lewis for his help during this research.

Furthermore, I would like to thank my colleagues José Serra, Pedro Mendes and all past PIC members. It was a great pleasure to work with all of them. I am grateful to José and Pedro for their comments and opinions. Besides, I would not obtain half of the results provided in this dissertation without Pedro's face. I am also thankful to Xenxo Alvarez, Hiroki Itokazu, Faceware Technologies and Jason Baskin for the models. Thanks also goes to Alexandra Ferreira for many administrative tasks she handled for me.

I especially thank my best friend Bayındır Saraçoglu for all his help and support.

I can never thank, Silvia, enough for her constant company and care that supported me through all the tough times.

Finally, I deeply thank my family for their unconditional love and support. Without them, none of this would have been possible.

This research has been supported by Fundação para a Ciência e Tecnologia/FCT (SFRH/BD/82477/2011), POPH/FSE program.

Abstract

Computer generated 3D facial animation, where humanoid or cartoon character faces are animated to express emotions, plays a tremendous role to enrich entertainment, perception, education and communication. Despite notable advances in the field of computer graphics addressing the 3D character animation, the evolution of the current tools and workflows of professional studios are still cumbersome to create the content for games and movies. As a result, producing high quality facial animation is a laborious, difficult and time-consuming process which requires both technical expertise and artistic talent. Animators widely focus on a set of control structure, which is called rig, to create the appealing expressions by manipulating the 3D virtual face. However, constructing facial animation using the rig requires deep understanding of the mechanics beneath that control structure and it involves long training periods and hard manual effort to carefully locate and manipulate the rig elements to create believable results. This dissertation describes novel methods to ease the facial animation workflow by proposing a set of *rig-focused* algorithms that are designed to create desired facial expressions practically.

Instead of considering the rig as a set of control structure, we approach the facial rig as an artistically designed deformation space to create the required facial deformations for each pose. Therefore, three topics, which lie in the rig deformation space, are investigated. The first method explored in this research addresses the direct manipulation of the facial rig structure using a sketch-based interface. The method allows to manipulate a large number of blendshapes, which is a widely adapted technique for high-quality rigging, simultaneously through a simple stroke in the form of free-hand drawing. Although there have been recent efforts on direct manipulation of blendshape models, the final results do not fully satisfy the artists. This is due to the fact that the mathematical frameworks of direct manipulation cause unintuitive and unexpected results with a global deformation impact. Our solution reduces the deformation impact to the local facial geometry by employing the geodesic circles,

which allows to gain intuitive and satisfactory results. To this end, we have analyzed the blendshape input data, which includes the vertex coordinates of the target shapes, and reconfigured the content of the input data according to the desired manipulation area. Our method is generic and responds automatically without any manual editing. Inspired by artists' brush painting on canvas, the method provides drawing strokes directly on the 3D model.

The second method proposed in the scope of this research is a novel transposition based approach for blendshape direct manipulation. The existing direct manipulation methods employ the pseudo-inverses for the underlying mathematical frameworks. Nevertheless, pseudo-inverse based approaches have a well known instability which causes unexpected facial movements during the artistic editing. Therefore, we present a novel and stable method, inspired by the *Jacobian Transpose Method*, to overcome the general instability problem of the blendshape direct manipulation with a pin-and-drag interface.

The third method deployed in this dissertation provides a novel Verlet integration framework which allows a local physically-based deformation on the arbitrary selected area of the facial model by using position-based constraints. Traditional frameworks deform the models as a whole. The proposed framework allows an interactive selection of the deformation influence area over the model, which improves computational performance for the position manipulation during the simulation. To this end, we take advantage of a geodesic distance computation technique based on heat kernel to determine the selected area. This approach allows producing life-like deformation effects by enhancing the animation especially during the collision handling phase.

The applications of the proposed methods to actual data are showcased by executing them to the production quality facial models. We validate our results by comparing them with the results of the existing methods. The comparisons are performed qualitatively and quantitatively that are demonstrated on graphical charts and side by side visual figures. Consequently, the proposed methods in this dissertation aim to reduce time and human effort during facial animation process by preserving the high quality results with stable and practical solutions.

Resumo

A animação facial 3D de personagens realistas e não realistas, gerada por computador, tem um papel crucial para enriquecer a experiência tanto a nível do entretenimento como da percepção, educação e comunicação, pois é através de movimentos faciais que as personagens exprimem emoções. Apesar dos avanços notáveis no campo de computação gráfica no que toca a animação de personagens 3D, o fluxo de trabalho é ainda moroso na fase da criação de conteúdo para jogos e filmes. Como resultado, criar animação facial de alta qualidade é um processo árduo, difícil e muito demorado que requer conhecimentos técnicos e artísticos. Os animadores tendem a focar-se na estrutura de controlo, chamada de *rig*, que é usada para manipular faces 3D, permitindo assim a criação de expressões apelativas. No entanto, criar uma animação utilizando o *rig* requer um conhecimento profundo das mecânicas da estrutura de controlo, o que envolve não só um longo período de formação, como de esforço manual sobre o modelo, para conseguir localizar e manipular os controlos do *rig* e assim criar resultados credíveis. Esta dissertação descreve métodos inovadores que facilitam a animação facial através de algoritmos focados no *rig* e que foram desenhados para permitir a criação de expressões faciais de forma prática.

Em vez de considerar o *rig* como uma parte da estrutura de controlo, nós abordamos o *rig* facial como um espaço de deformação para a criação das deformações necessárias à obtenção de cada pose. Assim sendo, foram investigados três tópicos ligados aos espaços de deformação de *rigs*. O primeiro método explorado aborda a manipulação directa da estrutura do *rig* facial através de uma interface baseada em esboços. Este método permite manipular um número elevado de *blendshapes*, técnica muito usada em *rigs* de alta qualidade, simultaneamente através do desenho de uma simples curva, tal como é feito em desenho livre. Embora existam esforços recentes no campo da manipulação directa de modelos com *blendshapes*, os resultados finais não satisfazem os artistas. Isto deve-se ao facto da *framework* matemática da manipulação directa causar resultados não intuitivos e inesperados com impacto global no modelo. A nossa

solução reduz o impacto da deformação a uma escala local através da utilização de círculos geodésicos, permitindo assim produzir resultados intuitivos e satisfatórios. Para tal, analisamos os dados da matriz de *blendshapes* que incluem as coordenadas dos vértices das poses alvo e reconfiguramos o seu conteúdo de acordo com a área de manipulação. O nosso método é genérico e responde automaticamente sem qualquer tipo de edição manual. Este método foi inspirado na forma como os artistas desenham numa tela, permitindo desenhar curvas directamente no modelo 3D.

O segundo método proposto no âmbito deste estudo é uma abordagem original baseada na transposição para a manipulação directa de *blendshapes*. Os métodos existentes usam a pseudo-inversa nas respectivas *frameworks* matemáticas. No entanto, abordagens baseadas na pseudo-inversa são reconhecidamente instáveis, pois causam movimentos faciais inesperados aquando da edição artística. Consequentemente, apresentamos um método novo e estável inspirado no *método de transposição Jacobiano*, superando assim este problema geral de instabilidade associado à manipulação directa de blendshapes que usam uma interface de *"pin and drag"*.

A terceira método proposta nesta dissertação é uma *framework* inovadora de integração de verlets, que permite deformações locais, baseadas em leis físicas, sobre uma área arbitrariamente seleccionada de um modelo facial, através da utilização de restrições *point-based*. *Frameworks* tradicionais deformam o modelo como um todo. A *framework* proposta permite a selecção interactiva da zona de influência da deformação no modelo, permitindo assim ganhos significativos a nível da eficiência computacional na manipulação das posições seleccionadas dos componentes primitivos da *mesh*. Para seleccionar a área utilizada do modelo, tiramos proveito da técnica de cálculo de distâncias geodésicas obtidas através de um *heat kernel*. Esta abordagem produz efeitos realistas, melhorando a animação especialmente na fase de colisões.

Os métodos propostos são demonstrados através da sua aplicação em modelos faciais com qualidade de produção. Os resultados são validados através da comparação com métodos existentes. As comparações são qualitativas e quantitativas, podendo ser visualizadas em gráficos e em imagens lado-a-lado. Os métodos propostos nesta dissertação têm como objectivo reduzir o tempo e o esforço associados ao processo de animação facial e permitir ao mesmo tempo manter os resultados de alta qualidade com soluções práticas e estáveis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Computer generated characters have recently become ubiquitous in movies, cartoons, advertisements and computer games. With the advances of technology and computer graphics techniques, it is currently possible to create lifelike characters. During the process of creating character animation, facial animation plays a key role to carry emotions and personality to the spectators. Especially, the increasing demand of the entertainment industry for high quality results drives the researchers to investigate new techniques to automate the animation workflow for realistic 3D characters. Despite the extensive use of 3D facial animation, the evolution of animation packages and workflows has been rather slow, because the adaptation of advanced methods to the current animation pipelines requires a long time period. Since production companies have their in-house tools and pipelines that are very well established over many years, accepting novel tools to be part of their pipelines requires a long term verification and training process. As a result, creating high-quality facial animation is still a labor intensive and time-consuming process. The goal of this dissertation is to propose a set of methods that speed up the human effort in the facial animation pipelines. We propose, firstly, a blendshape direct manipulation method localized on facial geometry with a user friendly sketch-based interface; secondly, a stable transposition based blendshape direct manipulation method; and finally, a novel solution to local deformation of the facial surface with position-based constraints. This chapter describes the general motivation, presents a brief summary of the proposed methods and contributions of this dissertation.

1.1 Motivation

Traditionally, facial animation pipeline is divided into three main stages: modeling, rigging and animation (see Figure 1.1). During the *modeling* stage, 3D facial models are created with a neutral pose as a static object. A skilled artist can create the 3D model with software packages or alternatively the model can be obtained by scanning an actor's face. 3D models are represented by well-organized surface mesh which consists of large number of primitive faces in the form of connected vertices. The facial mesh, as an end result of the modeling stage, is passed to the rigging stage to be set up with a series of controllers such as articulated joints, blendshapes and deformation elements to form the desired expression or deformation. This stage is crucial since the purpose of rigging controls is to change facial poses and create desired deformations on the mesh. These controls are, thereby, conducted directly with input parameters to determine the range of deformations and allow the user to create facial poses intuitively with high-level settings. This whole process is called "*rigging*" [OBP+12]. After the rigging stage, the mesh with rig is passed to the last stage which is *animation*. Animators perform prolonged movements by using the rig to pose the face over time according to the scope of the animation. This process is called keyframe animation. The whole movement of the face is broken into key poses over a certain amount of time to represent the sequence of the overall expression. All these three stages require artistic skills, technical knowledge and challenging human effort to gain the final animation output.

Techniques to create animation are focused on the interpolation of the vertex positions of the mesh for each time frame. The main purpose of the rig is to define the meaningful deformation space to create appealing facial expressions by altering the rig parameters for the keyframe animation process. The configuration of the rig and the geometric complexity of the mesh are other factors which directly affect the performance of the animators. When the details of the facial geometry and the rig complexity increase, manually creating facial poses for each frame of the animation sequence becomes impractical. Thereby, manipulating rig elements to gain the desired facial pose in the production environment is a difficult task for the animator. Nevertheless, there exist practical user interfaces, which provide rig manipulation with their high-level controls. The user interfaces associated to rig parameters partially ease the workload of the animators. For example, blendshape is a widely adopted technique for high-quality facial rigging [LAR+14]. Blendshape model forms a specific facial expression as a sum of linear combination of target faces that are obtained by either a skilled

artist manually sculpts from a neutral facial mesh or scanning a human face with various expressions. Most commercial animation and modeling software packages support blendshape models with their weight editing interfaces. These interfaces, nevertheless, allow the animators to perform the pose editing process only on external modules, where each pose is represented by a slider. Although many advanced techniques and tools exist for facial deformation and animation, meanwhile animators are surprisingly creating the appealing facial expressions by using the traditional slider based interfaces. The major reason can be explained as the proposed state-of-the-art techniques are difficult to adapt on the regularly used processes.

In this dissertation, we aim to overcome the considerable human effort in the *animation* stage of the production pipeline by simplifying the mesh deformation in three different perspectives. First, we explore a direct manipulation technique embedded in a sketch-based interface to control and manipulate the blendshapes intuitively. Second, we propose our novel and stable transposition based blendshape direct manipulation technique. Third, we present a localized physics-based facial surface deformation framework which provides the position based geometric constraints. Our main objective is:

to design practical tools and interfaces that provide high-level rig manipulation for model deformation in the scope of facial animation.

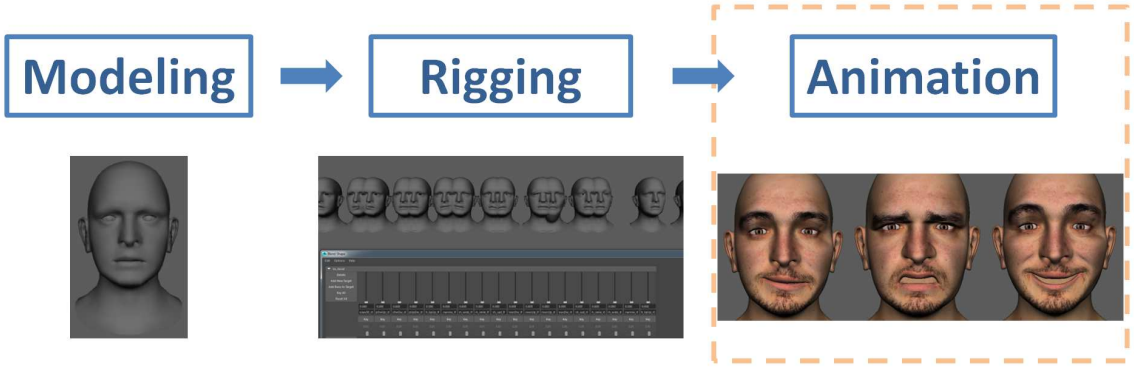

Figure 1.1: An overview of the main stages of facial animation pipeline. All stages are strictly connected to each other. Any possible alteration in one stage automatically affects the other stages. For example, a slight change of model geometry may cause a reprocess in rigging stage, or an additional facial pose request in animation stage may cause a significant change in both rigging and modeling stages. The scope of this dissertation targets the *animation* stage.

1.2 Overview and Challenges

This section summarizes the challenges and gives an overview of the three proposed methods that have been implemented to demonstrate the animation results. Since the foundations of the methods differ, the overall motivation shares the same goal that is to build user friendly animation tools.

1.2.1 Direct Manipulation of Blendshapes Using a Sketch-based Interface

In cartoon or classical animation, the process begins with artist's hand drawings of character' each pose on paper. This traditional "pen-paper" approach is considered as a powerful key element that allows the artists to design and control the pose of the character for each time-frame directly. Inspired from "artist's hand drawing" context, we propose a novel facial sketching control method that provides the rig manipulation of 3D facial models.

Recently, sketching tools have gained a significant value for creating, deforming or controlling the 3D models in computer animation. Most of the sketching methods have brought solutions to the problems which lie in vertex deformation space of the model. Our solution, nevertheless, targets the rig deformation space to manipulate the rig's structure beneath the model. Traditionally, animators determine the expression of the face indirectly by assigning values to each of the rig parameters individually. In order to gain a fully expressive deformation, animators have to handle hundreds of different rig parameters in a discontinuous approach. Our work attempts to bring the direct manipulation of blendshapes by using a sketch-based interface into animation work-flow for controlling the blendshapes in a continuous approach. The animator can manipulate several blendshape targets at the same time by using a simple stroke. This speeds up the creation of complex facial poses from multiple target shapes by only using simple hand movements. Despite the loss of precision, which is usually associated to the sketching interfaces, our method incorporates the common pin-and-drag interaction alongside the sketching for required fine-tuning.

Our method allows the animator to control a significant amount of blendshape targets without any training or understanding of the 3D model's inner mechanics in terms of rig. As a result, the animator can perform expeditious animation in real time by sketching strokes directly onto the 3D facial model.

1.2.2 Transposition Based Blendshape Direct Manipulation

The ability to directly manipulate the underlying rig of the facial model improves the controllability and reduces the animation production time by allowing the animator to alter multiple blendshapes simultaneously with a *pin-and-drag* operation on the surface of the model. This concept was originally inspired by *inverse kinematics approach*, and first formulated by Lewis and Anjyo [LA10] for the blendshape models. Their mathematical framework and the contiguous methods involve mostly pseudo-inverse based solutions. The major reason of pseudo-inverse usage is to compute a "best-fit" solution by following a least square approach. Especially in *inverse kinematics* applications, pseudo-inverses are heavily employed to find a solution of the end effector which moves a robot arm (in robotics) or a cartoon character leg (in animation), etc. As a matter of fact, there exists a close pseudo-inverse relationship between the manipulated points and blendshape weights in direct manipulation. However, pseudo-inverse is expensive to compute and has numerical instabilities [Wel93] due to spreading effect of matrix multiplication. The pseudo-inverse matrix columns, thereby, tend to have many non-zero values which produce many non-zero weights. Therefore, this instability of the pseudo-inverse causes to parametrize the least useful blendshape weights during the artistic editing of the facial model.

To this end, we develop a novel and stable transposition based blendshape direct manipulation method inspired by the *Jacobian Transpose Method*. We keep the classic *pin-and-drag* interface [LA10] to drive our proposed method. Our new method provides the artistic facial editing within the deformation directions and directly maps the pinned points' movements to the corresponding deformation directions. Our experiments show that our transposition based method produces more smooth and reliable facial poses than the existing pseudo-inverse based formulations. Besides, our method intuitively provides localized artistic editing without any additional parametrization or pre-computation.

1.2.3 Localized Verlet Integration Framework For Facial Models

Creating lifelike effects on the computer generated characters is a stand-alone challenge in animation movies or computer games. Besides posing the character by using the complex rig elements, preparing the virtual characters for the secondary motion effects such as collision, jiggling or bulging is also a vital problem. Physical phenomena is a

common approach to simulate the visco-elastic soft surface effects based on Newton's second law of motion (F=ma). The main purpose of the physics simulation is the integration of the 3D character to a virtual environment where external contacts are applicable. The most known technique for physics-based simulation is the mass-spring method with a proper time-integration scheme based on the networks of point masses on the mesh. However, geometric constraints based methods have proven their potential as an alternative to stiff springs with their stable time-integration. "Position Based Dynamics (PBD)" [MHHR07] and "Nucleus" [Sta09] are the popular works for the geometric constraints based methods.

Despite the advances in hardware performance, simulating soft surfaces is yet too expensive for complex models such as a face. Simulating 3D facial model's flesh by solving the equations of physics generates the behavior of real world elastic materials. However, achieving the result of one simulation step may take longer time than excepted. Existing frameworks and methods enable to perform the physics based simulations over the whole model. This approach is acceptable for the geometrically non-complex models or objects. However, performing a physics-based simulation on the surface of a facial model with thousands of vertices and edges demands a notable computing cost. In some particular cases, the animator is required to preview the result from simulation to control and fine-tune, hence the acceleration of the simulation process is inevitable. To this end, we present a framework which reduces the soft surface simulation from the whole model to an arbitrary selected area. The solver uses the Verlet time-integration scheme with a computation of the projected position constraints similar to PBD approach. Furthermore, we take advantage of the "heat method" [CWW13] for computing the geodesic distances to determine the desired deformation area. The framework is specifically designed for facial models. However, it can be easily integrated to other applications.

1.3 Main Contributions

The main contributions of this dissertation to the field of Computer Graphics are as follows:

- a **blendshape manipulator placing algorithm** to address the manipulator locating problem which is an open-ended and slightly unnatural task for the animators. In this novel algorithm, the animator simply sketches directly onto the

3D model the positions of the manipulators that they feel are needed to produce a particular facial expression. The manipulators activate the blendshapes in the model and allow the user to interactively create the desired facial poses by a dragging operation in screen coordinates.

- a **facial sketch-based control method** to manipulate multiple blendshape targets at the same time through drawing simple strokes directly onto the 3D model. The method allows the blendshape manipulation in a continuous way to create complex 3D facial deformations by using one single surface spline. The method employs the theoretical foundations of "Horizontal Curves" [Lev07] approach.

- a **localized blendshape direct manipulation** to give a solution to the global deformation impact of the original "direct manipulation blendshapes" method [LA10]. The problem lies in the "localization" of the pin-and-drag operations onto a 3D face model. According to the original formulation of [LA10], the operations affect the whole vertices of the face model. An efficient and effective localized operation is therefore desired to gain intuitive results. Our solution allows computing geodesic circles around the constraint points and limiting the edits to occur only within the circle.

- a **transposition based direct manipulation** to give a novel and stable solution to the instable pose editing of the original "direct manipulation blendshapes" method [LA10]. Our solution uses transposition based formulation instead of traditional pseudo-inverse based formulations. Besides, we provide an intuitive localized artistic editing during the facial pose production for keyframe animation.

- a **localized physics solver for facial animation** to enable the soft surface simulation on the arbitrary selected area of the 3D facial model. Traditional frameworks deform the models as a whole, however our Verlet integration framework takes the advantage of a state-of-the-art geodesic distance computation technique, heat method [CWW13], for the interactive selection of the deformation influence area over the model. Our solution gives a notable benefit in terms of performance for the position manipulation during the simulation process. Although the framework is specifically designed for the facial surfaces of the cartoon characters in 3D computer animation, it can be easily integrated to the existing frameworks.

1.4 Thesis Outline

The remaining chapters of this dissertation are organized as follows:

- **Chapter 2**: Describes the related work on blendshape models, facial deformation, sketch-based interfaces and posing, position-based dynamics approaches and soft surface simulation methods.
- **Chapter 3**: Describes the *Direct Manipulation of Blendshapes Using a Sketch-based Interface* method in detail. It presents the major problems faced and challenges to overcome during each stage of the method. After, it introduces the details of our solution to blendshape direct manipulation and the novel sketching interface algorithm developed for this research. The experimental results and analysis are further demonstrated in this chapter.
- **Chapter 4**: Describes the *Transposition Based Blendshape Direct Manipulation* method in detail. It presents the motivation of the main problem and describes the details of the proposed method. The comparisons and error analysis are further reported in this chapter.
- **Chapter 5**: Describes the *Localized Verlet Integration Framework For Facial Models* for simulating the deformation of soft surfaces on the arbitrary selected area of the 3D facial model in detail. It describes the algorithm we have developed to integrate the 3D face model into the virtual environment where external contacts can be applicable. Inspired by the position-based dynamics approach, the mathematical aspects of the position constraints are explained in this chapter. The deformation results are demonstrated on the various production quality facial models.
- **Chapter 6**: Concludes the thesis with a summary of our conclusions and discusses the potential future directions.

Chapter 2

Related Work

Creating appealing facial expressions is one of the most crucial steps in the character animation workflow. Animator's major goal is to manipulate the model's rig to create complex 3D deformations for bringing the character to life. When we consider that an individual character may contain several rig controls through the whole body, different deformation methods are required to create the proper animation. For example, the deformations for facial movements can be controlled by various techniques such as articulated joints, blendshapes or physics-based functions. Computer graphics' researchers have been working on these character deformation methods over many years. In this chapter, we present an overview of various deformation techniques that have been already in use for the facial animation process. Afterwards, we summarize the research about the sketch-based user interfaces that govern the hierarchy of the connected deformers and control parametrization. Next, the previous work on position-based dynamics methods will be reported. The chapter ends with a brief discussion on some open issues related to facial animation. The main purpose of this chapter is to give an overall understanding of the context that is related with the scope of the proposed dissertation.

2.1 Rig Based Deformation

Facial animation is a challenging problem for the computer graphics field over many years. In this section, we focus on the existing related work, which explored the rig based facial deformation to animate the digital face.

2.1.1 Skeleton Based Deformation

One of the most popular methods for facial deformation is to move the control handles of a skeleton hierarchy that is defined beneath the model. The working principle of this method starts in the rigging phase. A skilled artist locates the hierarchical skeleton graph beneath the rest pose of the model. This process has to be executed carefully, because the movements of the model vertices are dependent to the correspondent skeleton handles. Komatsu [Kom88] and Magnenat-Thalmann et. al. [MTPT88a] are prior examples of full body deformation controlled by articulated joints.

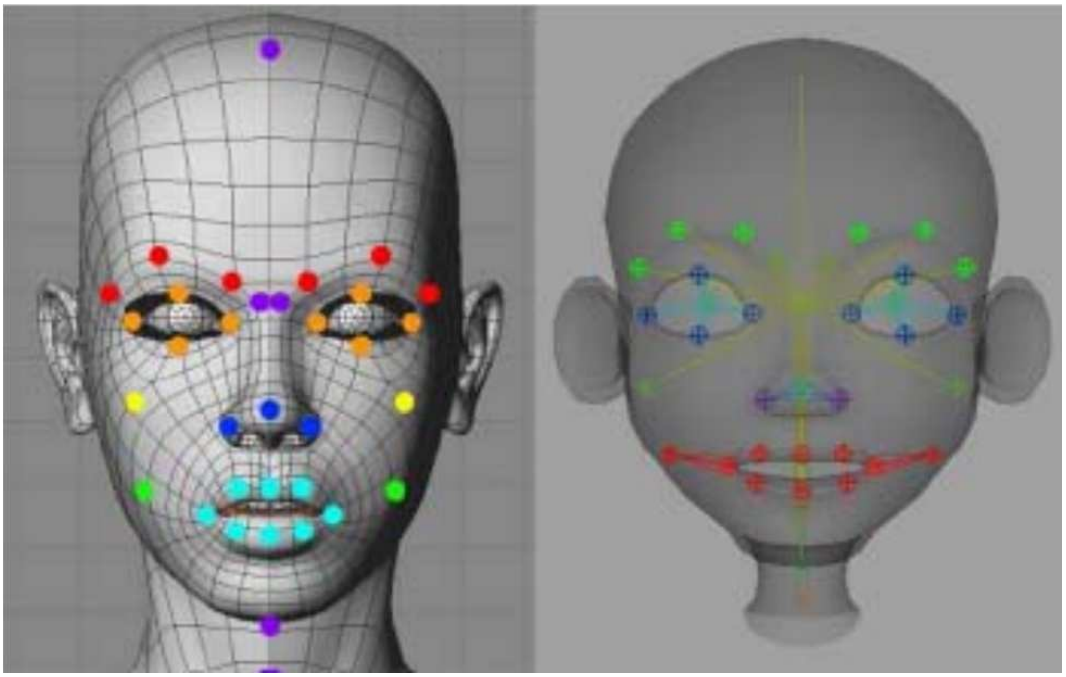

Figure 2.1: An example to facial rig based on skeleton hierarchy to control the facial movements and skin deformation [OMA09]

The connection of the articulated joints and the model geometry is a mandatory operation to animate the model. The general name of this connection is *skinning*. The deformation is controlled directly through joint-vertex influence weights (see figure

2.1). Easy usage and flexible adaptability throughout the softwares and hardwares make skinning beneficial in practice [WPP07]. There exist two types of skinning approaches: smooth skinning [YZ06] and rigid skinning [LCA05]. Defining weight is a crucial task for deformation of the skin because the influence impact of the vertex is highly dependent on weight definition [WP02]. Mohr and Gleicher [MG03] proposed an interactive authoring skin deformation by addressing the limitation of collapsing elbows. Lewis et al. [LCF00] introduced the pose space deformation approach that combines skeleton hierarchy and shape morphing. After, Kry et al. [KJP02] followed the pose space deformation by correcting the skeleton subspace deformation with vertex displacements. Additionally, James and Twigg [JT05] and Der et al. [DSP06] proposed another powerful approach that alternates to skeleton subspace deformation, which is called reduced deformable model. [JT05] proposed automatic algorithm to generate progressive skinning for articulated motion and [DSP06] offered a manipulation method, which provides direct control for the reduced deformable models. Baran and Popovic [BP07] offered an optimization method for the skeleton and skin weights coupling. Kavan et al. [KCZO07] offered another powerful optimization technique that blends the skin linearly using dual quaternions.

The general process starts to create sensitive skeleton hierarchies. The strategies on locating the skeleton structures beneath the model are reported in [McK06], [MS07], [MCC11a]. After the rigger fine tunes the bones for precise translation during movements of joints. As a third step, skinning process is prepared with the proper weight parametrization. Finally, testing of the manipulation is performed, this process explained in [RCB05] and [O'N08]. Besides, Orvalho et al. [OBP+12] reported the facial rigging techniques with deformation methods and recent trends. McLaughlin et al [MCC11b] surveyed an overview of many challenges that are involved in this area.

2.1.2 Blendshape Deformation

Blendshape is a popular approach for realistic facial animation and defined as "the simple linear model of facial expression" [LAR+14]. The known earlier works were realized by Fred Parke [Par72], [Par74], he experimented the linear interpolations between faces. With the work of Berggeron and Lachapelle [BL85], shape interpolation became popular. Similar to skeleton based approaches, blendshape animation starts with the creation of the blendshape. There are several methods exist in the literature. Blendshapes can be created by directly scanning the actor's face or a skilled artist sculpts the poses on a neutral model. There are several approaches for registering the

applied scans such as [LDSS99], [ARV07], [WAT+11], [ACP03] and [ASK+05]. Besides these methods, Pighin et al. [PHL+98] offered a method for fast creation of blendshape targets from multiple pictures of an actor. Blanz and Vetter [BV99] presented an approach on fitting a PCA model into a single picture and turning into actors face with close geometry and texture. Ma et al. [MJC+08] and Bickel et al. [BLB+08] presented their methods to create the fine details of the mesh such as wrinkles besides traditional normal and bump map methods. Finally, the most employed technique to create blendshape model is from a skilled artist manually deform a static neutral mesh into various target shapes according to the needs [LAR+14].

Producing animation with blendshapes requires defining weight values to blend the target shapes for each frame of animation (see figure 2.2). The techniques used to create animation can be briefly separated into keyframe, performance driven and direct manipulation. Blendshape models are conventionally animated by using the keyframe technique with the advanced support of the commercial software packages. Osipa [Osi07] proposed a manual for this broad technique. The performance driven methods focus on the motion of the actor for driving the facial model [Wil90], [CDB02], [BBPV03]. Although the keyframe animation methods are popular in animation movies, the performance driven technique are commonly employed in the visual effects of the movies [PF05], [PF06]. Beeler et al. [BHB+11] proposed a performance driven facial capture system by using geometry acquisition based on stereoscopic reconstruction. Huang et al. [HCTW11] introduced a blendshape interpolation framework, which reconstructs 3D facial performances by combining captured data with a minimum facial scans. Weise et al. [WBLP11] presented a practical method for animating face based on a tracking algorithm, which combines three dimensional geometry and two dimensional texture to gain motion to blendshapes. The major classification of the performance driven methods lies on the use of 3D motion captured data as an input [DCFN06], [CLK01]. Besides another classification includes methods that execute a model-based tracking of the particular video [BBPV03], [CXH03], [RHKK11], [BGY+13], [CWLZ13]. Performance driven methods often report the comparisons of PCA based approaches [BBPV03] and blendshape based approaches [CLK01], [CDB02], [CK05a].

Besides traditional keyframe and performance driven techniques, inverse kinematics based approaches [Wel93] have been employed to deform or pose character faces. These types of methods are also called *direct manipulation* methods. In these approaches, instead of editting the weight parameters individually, animator moves points or particular handles to control the model by solving an inverse problem to determine

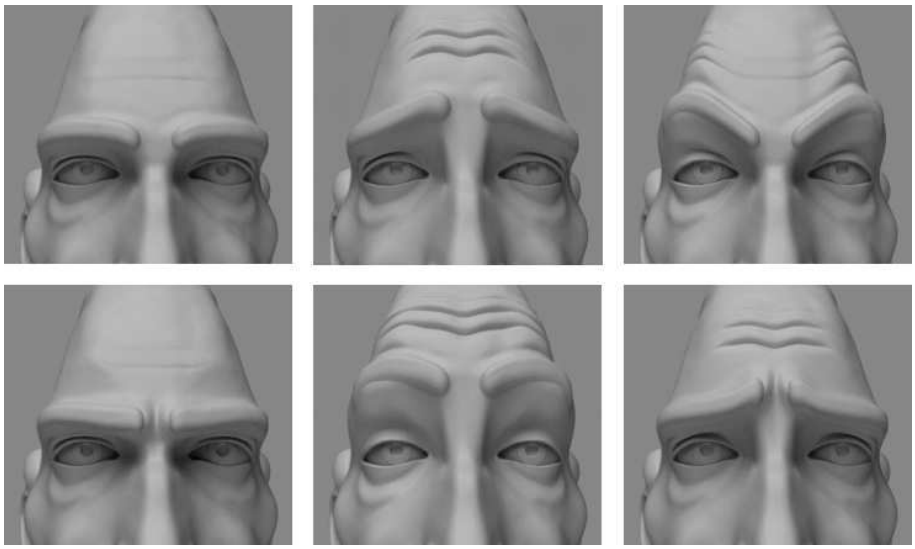

Figure 2.2: An example for different target shapes blend to create appealing facial expressions. Schleifer et al. [SSMCP02]

the underlying weights. However, solving the inverse problem is a challenging task for the direct manipulation blendshapes, because the number of constraints is high. In a production quality facial model, the number of blendshape targets can reach to hundreds. Therefore, finding a solution to the inverse problem of direct manipulation of blendshapes requires to find a discrete function, which satisfies all these constraints by touching and moving points on the surface of the model. At the same time, solver has to find the beneath weights that are most appropriate for that motion. Pighin et al. [PHL+98] presented a set of techniques, which allows to construct the textured target shapes rapidly from pictures of an actor with a painterly interface to blend the parts of the target shapes. Joshi et al. [JTDP03] proposed an automatic segmentation for localized blendshapes with a direct manipulation technique. Lewis et al. [LA10] developed a solution to this problem by formulating direct manipulation as a soft constrained inverse problem of finding the weights of blendshape for the selected point movements. The solution includes a regularizer for the most optimum results during interactive editing (see figure 2.3).

Seo et al. [SILN11] proposed an improved version of the regularizer term of the inverse problem and presented a new matrix compression scheme for complex models based on hierarchically semi-separable representations with an interactive GPU-based animation. Tena et al. [TDlTM11] offered to handle more local modification by dividing the PCA based facial model into independently trained regions, which share the boundaries with each other. This method also allows the user to directly interact

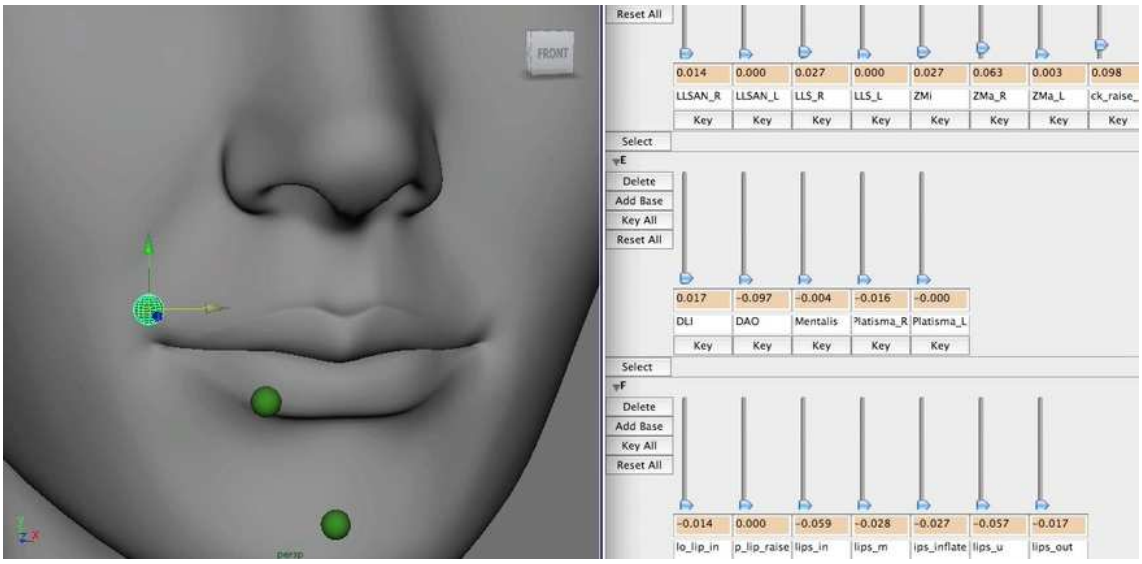

Figure 2.3: An example for direct manipulation blendshapes method to pin and drag the point on the surface of the facial model to determine the corresponding weights. Lewis and Anjyo. [LA10] and Lewis et al. [LAR+14]

with the model by pinning and dragging. Anjyo and Lewis [ATL12] presented an algorithm, which allows rapid and better edits that learned from the previous animation. In more recent work, Neumann et al. [NVW+13] introduced a method about localized deformation, which decomposes the global deformations to localized components from the performance capture sequences using a variant of sparse PCA. Yu et al. [YL14] proposed a method to reproduce the facial poses based on blendshape regression, which uses an optimization procedure to improve the quality of the facial expressions. Holden et al. [HSK15] developed a real-time solution to the inverse problem of rig function, which enables mapping of the animation data from character rig to the skeleton of the character. That method can be applicable to whole character animation as well as facial bone based animation. Our preliminary mathematical framework has followed the direction of [LA10] and we have performed the blendshape direct manipulation with an approach to locate the manipulators onto the facial model intuitively by using a sketch-based interface and apply the dragging operation after the sketching procedure (see figure 2.4). We refer to Lewis et al. [LAR+14] as a comprehensive report about practical and theoretical aspects of blendshape facial models.

Before the direct manipulation applied on blendshape models, it had been executed on the PCA based models. One of the first application in this field was seen in the work of Zhang et al. [ZSCS04]. They offered a method that obtains the target shapes from a high quality stereoscopic data on a template mesh and enable direct editing by using

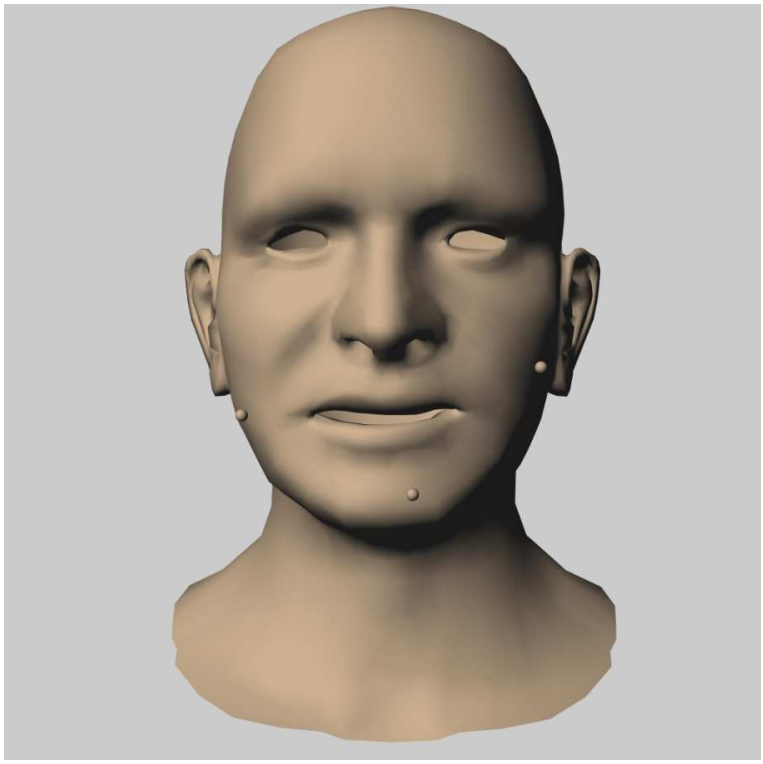

Figure 2.4: An example facial pose that is created by using our sketch-based controllers method. The results are promising in the early stage of the proposed research. [COL15]

adaptive local radial basis blends. Zhang et al. [ZLG+06] proposed a geometry-driven facial expression synthesis system, which supports hierarchical segmented PCA model during the point manipulation all hierarchy is effected from the movement. Meyer and Anderson [MA07] presented a key point subspace acceleration and soft caching method to use examples to compute a subspace and set critical points and then focus only these points during further computations. Li and Deng [LD08] extended the method of [ZLG+06] by using local and hierarchical PCA facial model to edit on basis of constrained weight propagation. Although the theory behind PCA and blendshape models are similar, the practical applications for direct manipulation show differences in terms of the level of detail to reach during the production. Direct manipulation approaches are practical for both experienced and novice users to reduce the intensive human effort to create facial poses [LA10].

2.1.3 Physics-based Deformation

Physics-based deformation techniques simulate the flesh, skin surface or muscles to gain life-like deformation. Simulating 3D facial model's flesh and tissues allows to

achieve more realistic animations and expressions by solving the equations of physics. The methods for simulating physics can be listed as mass-spring network systems, vector representations and layered spring meshes.

The early work was from Badler and Platt [BP81] which used mass-spring system to create a physics-based model. Waters [Wat87] proposed several muscle models consist of linear, sphincter and sheet effects independent from the skeleton hierarchy to drive the skin deformations. After, Waters and Terzopoulos [WT92] improved the previous work by adding three layers structure such as cutaneous tissue, subcutaneous tissue and muscle to create a realistic behavior with various spring parametrization. Magnenat-Thalmann et al. [MTPT88b] presented an abstract procedure for muscle activations in facial models. Lee et al. [LTW95] scanned the human face and used the data to simulate the muscle contractions. Waters and Frisbee [WF95] presented a speech engine based on simulation of muscles that shows effect only on the surface of the face.

Regrading the performance captured data extraction, Essa et al. [EBDP96], [EP97] proposed a system, which analyzes the facial motions with optical flow and combines them with geometric information and physics-based models to extract a parametric muscle group representation. Basu et al. [BOP98] explained a method based on tracking the human lip motions from a video sequence. The offered physically-based lip model was trained lip motions and PCA was employed to reduce the degrees of freedom, which allows to match human lip motions automatically. Choe and Ko [CK05a] simulated the actuation of muscles underneath the skin by using 2D linear quasi-static finite elements method and synthesize the facial expressions by a captured data.

With the hardware advances, the real-time physics-based muscle animation was investigated by Kahler et al. [KHS01], [KHYS02]. The proposed method was developed to demonstrate skin deformations based on muscle contractions and three layer model was built to simulate bulging caused by elastic behavior of muscles. In their model, mass-spring system, which used for muscles simulations connects the skull, muscle and other skin layers. Sifakis et al. [SNF05], [SSRMF06] presented an accurate and precise muscle-based facial animation system, which is using a non-linear finite elements approach. In their system, muscles are activated by sparse markers and serve as actuators over the skin surface. The system computes the environmental interactions such as collusion handling and modifies the animation behavior. Aina and Zhang [AZ10] proposed automatic rig-builder that generates soft-tissue layers and bone structure. (see figure 2.5)

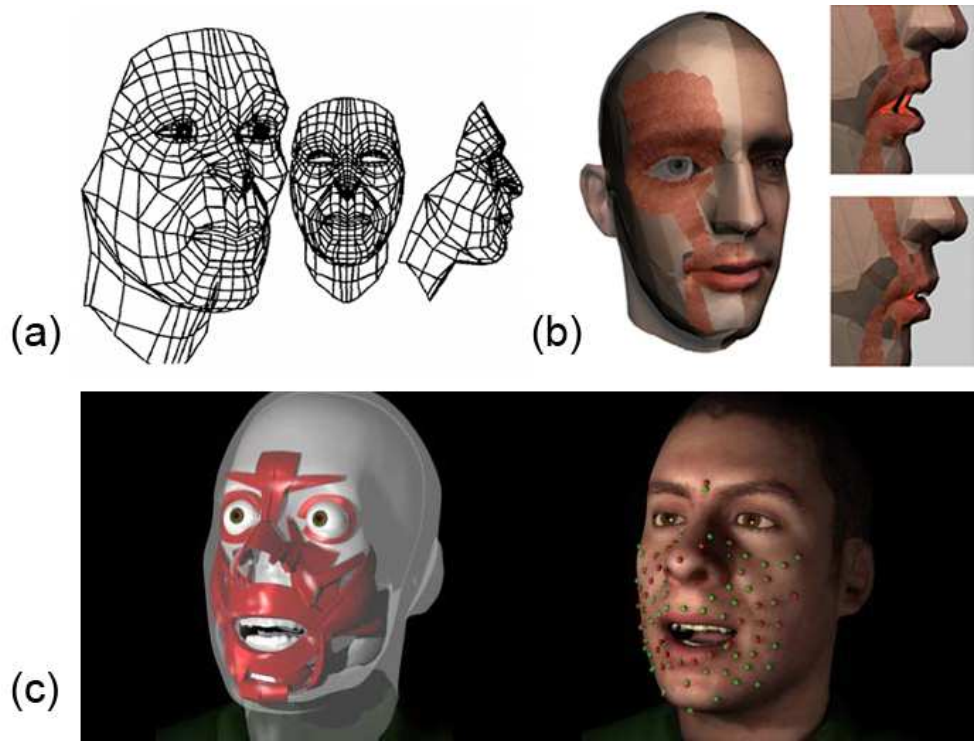

Figure 2.5: Examples of physics-based approaches for rigging and deformation. (a) prior mass-spring network for the facial muscles [BP81]; (b) facial muscle model with several layers [KHS01]; (c) Facial muscle and flesh simulation using finite elements approach [SNF05]

Besides the muscle simulations, physics-based skinning has increased its popularity for creating secondary motion effects during continuous movements of the characters. Although physics-based skinning approaches create a life-like visual results with an automatic calculation, the cost of the computations is considerably expensive, which limits the usage in interactive applications. However, the results of the recent researches in this subject are promising. Sueda et al. [SKP08] proposed a method for hand and forearm simulation by adding complex routing constraints on muscles and tendons. Kim and James [KJ11] proposed a domain-decomposition method for simulating the skin deformation within a subspace framework. The method works in interactive rates by supporting quasi-static and dynamic deformation with nonlinear kinematics. McAdams et al. [MZS+11] presented a novel algorithm for the skeleton driven, high resolution elasticity simulation. Algorithm performs near-interactive rates, which is designed for soft tissue deformation for virtual characters. Theory behind their algorithm is based on uniform hexahedral lattice, one point quadrature scheme and multi-grid method. One of the most efficient use of multi-grid method can be seen in Zhu et al. [ZSTB10]. They introduced a multi-grid framework for the simulation of elastic deformable models with an efficient parallel computation. Sin et

al. [SSB13] developed Vega non-linear finite element library, which supports several finite element methods to deform the models with different integration schemes. The library offers an efficient solution for physically-based simulation of the soft surfaces with its highly optimized algorithm and it can be extended to the existing frameworks easily. Cong et al [CBE+15] offered a fast and fully automatic morphing algorithm for generating physically based rig for human and humanoid faces. This algorithm is the most advanced physically based rigging to simulate flesh and muscle models. It only accepts a target surface mesh as input and the rest of the whole operation is executed without any user interaction. The final model can generate several expressions, which can be applied on different characters using the same muscle stimulation.

Hahn et al. [HMT+12], [HTC+13] presented secondary skin effects inside the span of the rig space, which allows the simulation of flesh and muscles in a nonlinear space. Besides, Lie et al. [LYWG13] offered a framework that enables coupling of skeleton and soft tissue dynamics for control and simulation of soft-body characters (see figure 2.6). Gao et al. [GMS14] introduced an efficient method that simulates flesh and supports an ex-rotated elasticity material model. This model offers a plausible approximation to corotational elasticity with a force model and pose-dependent coefficients. We refer to [RF16] as a comprehensive report about skinning techniques for articulated deformable characters.

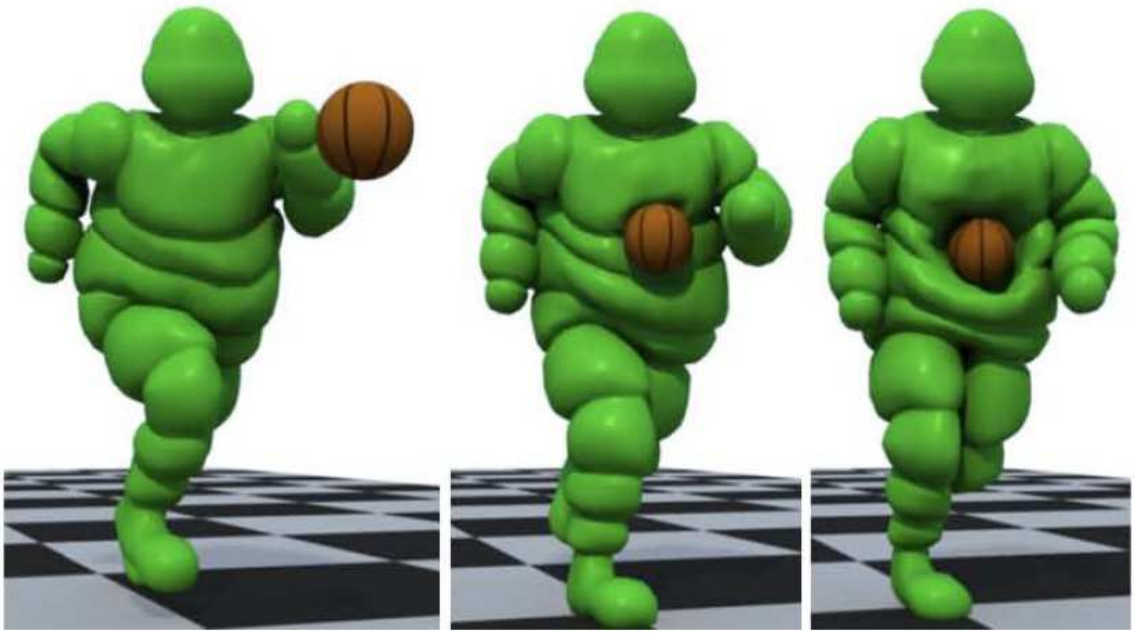

Figure 2.6: Example of a physics-based skinning with a skeleton and skin geometry coupling and allows collision handling. Liu et al. [LYWG13]

2.1.4 Geometric Deformation

Geometric deformation in rigging is a practical operation for the manipulation of the geometrically complex models. The operators used for geometric deformation provide simple and efficient interfaces to realize the deformation.

Free Form deformation (FFD) is considered as one of the prior work on geometric manipulation. It was first introduced by Sederberg and Parry [SP86] and is a commonly used operator, which deforms the volumetric object models with manipulation points that provided in a 3D cubic lattice. Chadwick et al. [CHP89] used FFD for the animation of hierarchically multi-layered character to provide more control to the animators. The layers in the hierarchy are listed as behavior, skeleton, muscle, fat layers with a surface description. Coquillart [Coq90] proposed extended FFD (EFFD) to provide more general lattices. Kalra et al. [KMMTT92] presented Rational Free-Form Deformers (RFFD) by extending FFD to control the animation. This method incorporates weight parameters for each control point, which provides better control over the geometry. Another important point in this method, Abstract Muscle Action (AMA) procedure is used for controling the facial animation. AMA procedure was introduced by Magnenat-Thalmann et al. [MTPT88a], which simulates the specific action of a face muscle. Each AMA procedure represents a group of muscle that are used to form the facial expressions. Noh et al. [NFN00] used RFFD for creating smooth animation procedure on facial models. Luo and Gabrilova [LG06] improved FFD by using Dirichlet-Voronoi diagrams based data interpolation, which removes constrains on lattice control. This method is called Dirichlet Free-Form Deformers (DFFD) and is practical to model facial models from using a pair of input images by using control points.(see figure 2.7)

Besides FFD, Signh and Fiume [SF98a] proposed to shape the model using wires. The wires are taken as parametric curves and control the appearance of a region of the model. This method can be used to create various poses with its parameters. Later, Singh and Kokkevis [SK00] offered a skinning technique in combination of FFD for 3D character animation.

D.O.G.M.A. model (Deformation Of Geometrical Model Animated) was proposed by Dubreuil and Bechmann [DB96] for facial animation. This method allows to define displacement constrains for space deformations and deformation can be governed over time. Finally, Sumner et al. [SZGP05] introduced mesh-based inverse kinematics (MeshIK), which offers a novel algorithm for posing meshes without any rig control or parameter. This method allows to manipulate complex models by using a few

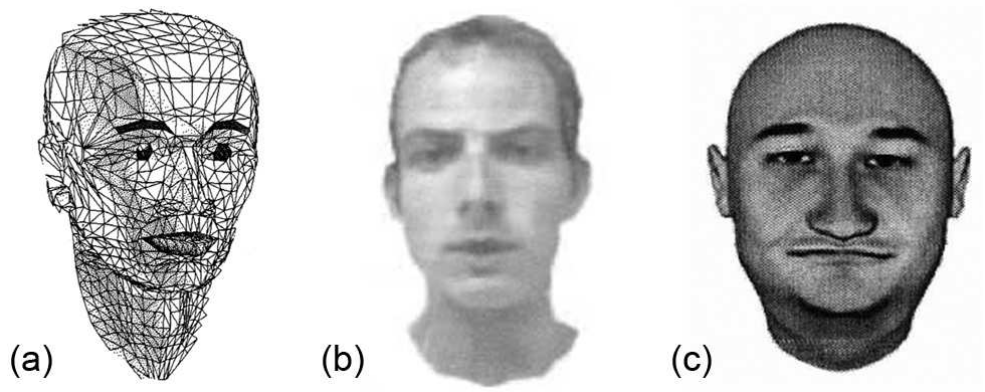

Figure 2.7: Examples of geometric deformations. (a) facial animation with D.O.G.M.A. model Dubreuil and Bechmann [DB96]; (b) Rational Free-Form Deformers (RFFD) for face model Noh et al. [NFN00]; (c) Dirichlet Free-Form Deformers (DFFD) applied on face [LG06]

points and method produces plausible deformation automatically. Later, Sumner et al. [SSP07] proposed another method for wide range of shape representations and editing scenarios using a collection of affine transformation organized in a graph structure.

2.2 Sketch-based User Interfaces

3D modeling and animation is a laborious and time consuming task for animators because of its geometrical complexity, huge number of various shapes involvement and their relationship in the scene. Current professional software packages feature powerful and fast tools with advanced solvers for detailed modeling and manipulation. However, these packages only offer standard interfaces such as window, icon, menu, pointer (WIMP) [OSSJ08] to perform the operations. Researchers, thereby, have been investigating on sketching interfaces as an alternative to simplify the cumbersome interaction procedure. Inspired from cartoon drawings, the main purpose of the sketching interfaces is to speed up the process of creating or manipulating 3D models in the animation workflow. This concept is also called as Sketch-Based Interfaces for Modeling (SBIM), which allows to converge various modeling modules and create a stable integration with the traditional external interfaces in a natural way.

The early research on sketching was realized by Sutherland [Sut64]. The proposed system was named as SketchPad, which provided the direct manipulation of the objects from the screen by using a light-pen. After, sketch-based methods used in a wide range of applications such as 3D design and modeling [DJW+06], [ASSJ06], animation

[CON05], engineering [VMS05], deformation [ESA07] etc. We refer to Olsen et al. [OSSJ08] as a comprehensive report about the modeling techniques using sketch-based interfaces.

According to the report of Olsen et al. [OSSJ08], the pipeline of general sketch-based systems are divided into 3 subsequent stages. Sketch acquisition from the user's action, is the first stage and followed by sketch filtering, which transforms the raw sketch to a usable state and last stage is sketch interpretation, which enables to operate on 3D model with the sketch. In our sketch-based interface, we have taken the advantage of the general pipeline and applied all the stages to our interface in the order of acquisition, filtering and interpretation. The following subsections detail the stages of the sketching pipeline.

2.2.1 Acquisition

The concept of sketching starts by obtaining the drawn stroke from the user. With the advances on the tablet and mobile devices in the last decade, current technology allows us to directly interact with the displays. For example user can draw, play or write on the display just using the finger tips. In past, nevertheless, this type of interaction was possible only by using special purpose input devices such as wire connected pens. Although the advanced mobile devices allows natural direct interaction, the desktop computers are still acquiring the input through standard keyboards and mouses. However, traditional mouse is a sufficient device to mimic the natural free-hand drawing on the screen of desktop computer through high-end software packages.

There are several methods that exist to transfer the pen and paper drawing context to digital environment. Haptic equipments are effective in terms of providing accurate feedback through their special tools. Hayward et al. [HACH+04] reported the practical usage of haptic devices with their interfaces in detail. Besides haptics, responsive workbenches or interactive tabletop displays are valuable alternatives. Schmidt et al. [SWSJ06] introduced ShapeShop sketch-based modeling system, which works in a responsive workbench by using BlobTree framework. This framework allows to create complex, detailed solid models with arbitrary topology. Fleisch et al. [FBSS04] showed that sketch-based techniques are also compatible with 3D virtual tools.

Recognition of user drawn stroke is the main challenge to operate with the limited information provided by sketching. Sezgin et al. [SSD06] implemented a system, which is a combination of several sources of knowledge to provide the early processing

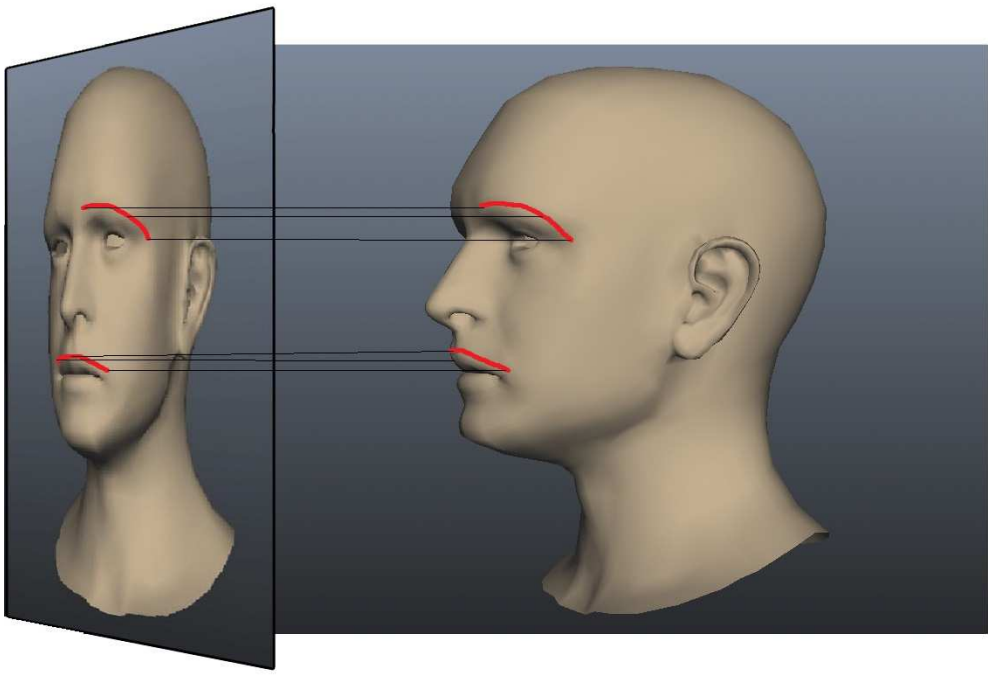

Figure 2.8: Example of the transformation of a given 2D stroke to 3D curve by using the canvas. [COL15]

for freehand strokes. In the proposed system, they defined the pixels from the drawn stroke and created low level geometric simple objects such as lines, curves, rectangles, ovals, polylines etc.

Another style of recognition is the concept of "drawing on canvas". Das. et al. [DDGG05] used the canvas to transform the sketch from a 2D drawing plane to 3D world coordinates. They proposed to interpret 2D curve to be the projection of 3D curve. The canvas is also popular in facial modeling and animation, which will be discussed later in this chapter. Tsang et al. [TBSR04] presented a sketching tool for 3D wireframe models, which has an image guided pen-based interface. In this tool, canvas replaced by images to guide the user. Yang et al. [YSvdP05] offered a sketching system for modeling with its novel algorithm that searches to match the points and curves of a set of given 2D templates to the sketches. The system employs an optimization metric for the matching process. In our sketching user-interface, we employ the *canvas concept*. The freehand stroke is applied to the imaginary canvas, which is aligned to the screen space. Afterwards, 2D stroke is transformed to the 3D

spline curve (see figure 2.8). Further details of our sketching algorithm is presented in Chapter 3.

2.2.2 Filtering

Filtering is a necessary step before performing an interpretation of the sketch to clean the noise from the given stroke. Sezgin and Davis [SD07] proposed a method for threshold-free feature point detection using scale-space theory, which focuses on the user's poor drawing skills and device errors. Biermann et al. [BMZB02] introduced a sketching method for generating sharp features and trim regions on subdivision surfaces. Araujo and Jorge [AAJ03] developed a software called BlobMaker for free form surface modeling, which defines various implicit surfaces geometrically to edit objects using sketches. Kara and Shimada [KS06] offered another software that is based on sketching for designing 3D objects. Their tool works in 2 stages: at first stage, user designs a wireframe model and in second stage user builds surface interpolation on the wireframe to gain the final solid model. Olsen et al. [OSS07] presented a new optimization approach to angle quantization for stroke's each point elements. All aforementioned methods can be categorized as resampling filtering against the drawing speed of the user. The extreme type of resampling is polyline approximation, which is an efficient way to reduce the noise of the given stroke. Igarashi et al. [IMT99] used the polyline approximation in their software package, called Teddy, which creates plausible 3D polygonal surfaces from drawn 2D freehand strokes (see figure 2.9). Kurozumi and Davis [KD81] presented the minimax method, which minimizes maximum distance between an arbitrary point and the approximated line for the problem of approximating digitized picture by polygons. Saykol et al. [SGsGU02] used another polygonal approximation approach with a kinematics based method. They simplify the polygon vertices according to their velocity and accelerations with respect to the center of the polygon.

Curve fitting is another powerful approach to reduce the approximation errors. There are several methods used such as polynomial fitting with least square approximation, which was explained in [Koe98], Bézier forms [Pie87] and B-splines [KS06]. Furthermore, Alexe et al. [AGB04] presented an interactive modeling technique, which uses Haar wavelet transformation to obtain the strokes. Cherlin et al. [CSSJ05] applied revers subdivision to raw stroke samples for fitting a curve to a stroke. Sketched curve may include linear and smooth sections, thereby segmentation is another alternative for fitting the curve to the stroke [SMA99], [KGG02], [SSD06], [SMA99].

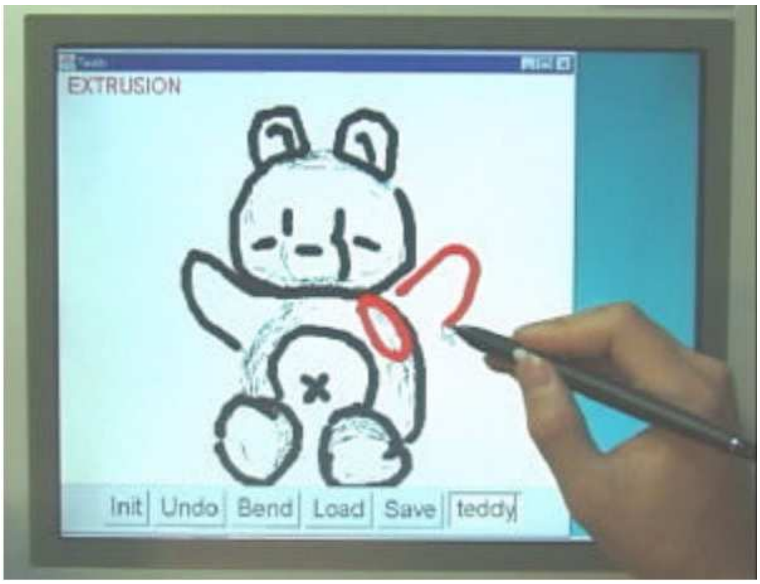

Figure 2.9: The screen shot of Teddy software package for modeling objects through sketching interface. Igarashi et al. [IMT99]

User can always miss the drawing, so redo operation for sketching is inevitable. It is beneficial to expect the over drawing updates from the sketching systems. This concept is called oversketching and can be seen in [FRSS04], [AAJ03], [SSD06]. Besides in some other systems such as [CSSJ05], [KS06], [KDS06], [NISA07], [WPP07], the original sketch is kept and they allow for oversketching at a later time. On the other hand, some systems blend multi strokes together. Pusch et al. [PSNW07] analyzed multiple strokes and replaced them with a single stroke that is the most appropriate estimation for the artist. Kara and Stahovic [KS05] proposed an approach for stroke blending targeting the images. Aesthetic is an important factor during modeling and beautification [IMKT97] is expected from the sketch based system. Igarashi et al. [IMKT97] first revealed the beautification and applied interactively. Igarashi and Hughes [IH01] introduced an interface for 3D modeling, which also contains the interactive beautification. Contero et al. [CNJC03] developed the CIGRO system, which provides a simple calligraphic interface for generating polyhedral objects.

2.2.3 Interpretation

After the given sketch is filtered, the system interprets the sketch according to the target operation. In Olsen et al. [OSSJ08], several operations are categorized for sketching interpretation. These operations are listed as automatic 3D model creation, augmenting the existing models, deforming the models, representing the surfaces and

designing the interface.

Creating 3D models using a sketch based system requires identifying 3D object from a 2D drawing area. According to Olsen et al. [OSSJ08], creation of 3D models can be considered in two contexts: suggestive and constructive. There 2 phases in suggestive systems, first is the recognition of the sketching template and after using that template for generating the model. Contrary to suggestive systems, constructive systems skip the recognition and directly perform the object modeling. Suggestive sketching is limited with the memory capacity because of the recognition phase. Zeleznik et al. [ZHH06] proposed an interface called SKETCH, which uses non-photorealistic rendering and gestural interface for simple line drawings of primitives. Pereira et al. [PJBF00] introduced another gesture based modeling system called GIDeS with an advanced memory features to allow the user to draw on the previous modeling work. Both systems extrapolate the final model based on strokes. Igarashi and Hughes [IH01] also takes the advantage of extrapolation. Funkhouser et al. [FMK+03] offered a search engine based on shape-based search methods. This web-based system supports queries based on 2D-3D sketches with 3D models and text based keywords. Shin and Igarashi [SI07] proposed a system with a model database called Magic Canvas, which takes simple 2D strokes as input and performs an automatic identification for the corresponding matches in the database (see figure 2.10). Constructive sketching is another alternative for creating 3D models without recognition step. Pugh [Pug92] introduced a mechanical solid modeling system called Viking, which is an interactive sketching interpretation and provides a 3D reference with geometric constraint definition explicitly. Other systems such as [PJBF00] and [CNJC03] are also compatible for interactive mechanical solid design with their visual feedback feature. Varley et al. [VMS05] proposed a method that inflates a drawn sketch to generate the frontal geometry and determines the existence of T-junction. Masry and Lipson [ML07] presented a computer aided design interface, which performs an optimized algorithm to reconstruct sketches of 3D objects from lines and planar curves. Varley et al. [VTMS04] proposed a system, which enables to draw the structures with straight lines and the system converts the drawing with curved segments.

According to Hoffman [D.D00], human visual system can interpret the line drawings as 3D contours since it is possible. Interpreting drawn strokes as contour lines is a common idea in sketching research. Eggli et al. [EHBE97] described a pen based quick sketching approach for 2D and 3D modeling, which uses the sketch to define exact shape's geometry by using sketch input. This approach is one of the first examples of contour lines. After, [SC04], [SWSJ06] and [WPP07] applied the concept of contour

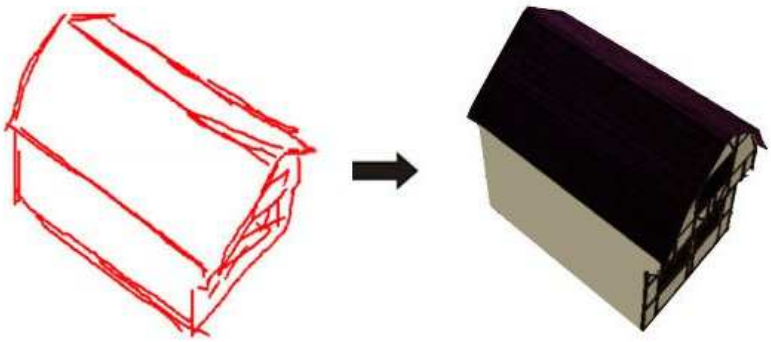

Figure 2.10: Magic Canvas system is an example for the suggestive sketching with its automatic recognition feature from its database. Shin and Igarashi [SI07]

lines in their methods to create object surfaces with linear sweep technique. For obtaining smoother cross-sections of the object surfaces Cherlin et al. [CSSJ05] used the medial axis of two strokes to create a generalized surface. Besides, Teddy system from Igarashi et al. [IMT99] computes the Delaunay triangulation [DB00], creates polygon approximation, extracts chordal axis, inflates the objects by pushing vertices for inflating closed contours. Another valuable approach is the skeleton of the closed contour, Alexe et al. [AGB04] used a spherical implicit primitives for each skeleton vertex. During the blend of primitives, it generates a smooth surface. Karpenko and Hughes [KH06] introduced SmoothSketch system, which uses again visible contour sketches to obtain visually plausible 3D objects. Cordier and Seo [CS07] offered an algorithm, which understands the hidden parts of the contour and generates a smooth 3D object from these contours by giving a solution to a set of optimization problems. Nealen et al. [NISA07] also used a nonlinear optimization technique for smoother surfaces and sharp edges (see figure 2.11).

The object boundaries are another important factor for interpretation. Das et al. [DDGG05] defined the boundaries by using 3D network of curves and used a smooth interpolation for building the objects. Rose et al. [RSW+07] developed an algorithm that generates developable surfaces by using a given 3D polyline boundary sketches.

Giving details to an existing 3D object is relatively an easier task compared to creating 3D objects by using 2D sketches, since the model serves in the scene with 3D destination coordinates for mapping the given stroke into the 3D coordinates (see

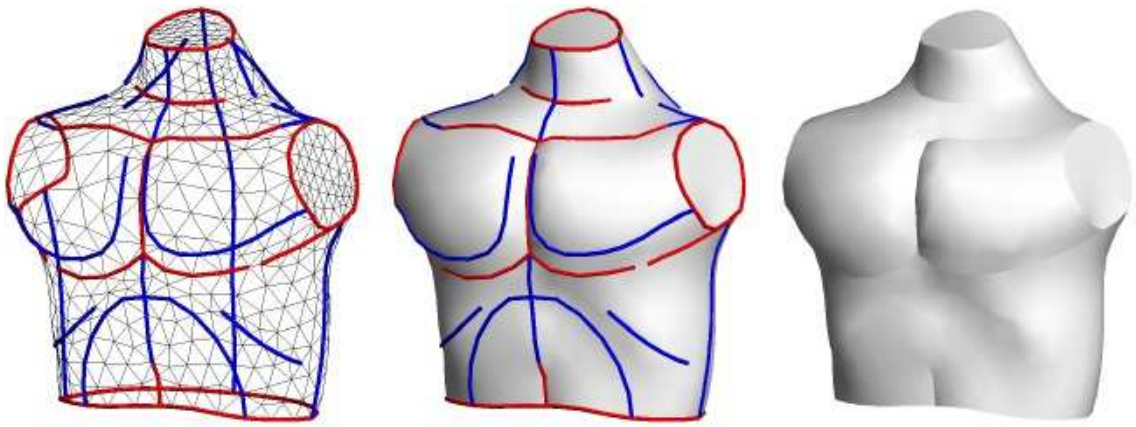

Figure 2.11: Example image of result from FiberMesh for smooth surface creation with strokes. Nealen et al. [NISA07]

figure 2.8). There are several techniques for mapping the stroke onto the 3D model such as [OSSJ05], however the most common technique is ray casting (rays shoot from the stroke directly to the 3D model). According to Olsen et al. [OSSJ08] augmentation can be grouped into two sets: surficial and additive. The surficial augmentation is used to sketch features to give arbitrary details to the model's surface. The examples can be seen in [BMZB02] or [OSSJ05]. Besides, given strokes may add new geometric constraints to the surfaces [NISA07]. Beneath surface representation remains often same in surficial augmentation [NISA07], [OSSJ05]. The additive augmentation is used to define new parts to the model such as limb by using constructive strokes. Some approaches use additional strokes for the connection of the new parts to the main model [NISA07], [IMT99]. Schmidt et al [SWSJ06] enabled the additive augmentation by using easy blending with an implicit surface representation and their system executes a parametrization for controlling the smoothness at connection region.

Surface deformation is another popular approach in sketch-based systems because of its intuitive and simple interpretation. Cutting the model was applied in [WFJ+05], [JLCW06]. Wywill et al. [WFJ+05] implemented the cutting as an additional feature to their shape editing tool and allows the user to create holes or removing volumes directly with the cutting feature (see figure 2.12). Ji et al. [JLCW06] proposed an intuitive tool called Easy Mesh Cutting, which allows the user to cut arbitrary components from the object by using freehand sketches on the mesh. Bending and Twisting deformations can be applied by using the oversketching concept [NSACO05], [ZNA07].

In oversketching the main idea is using 2 strokes, one is for reference and the other is

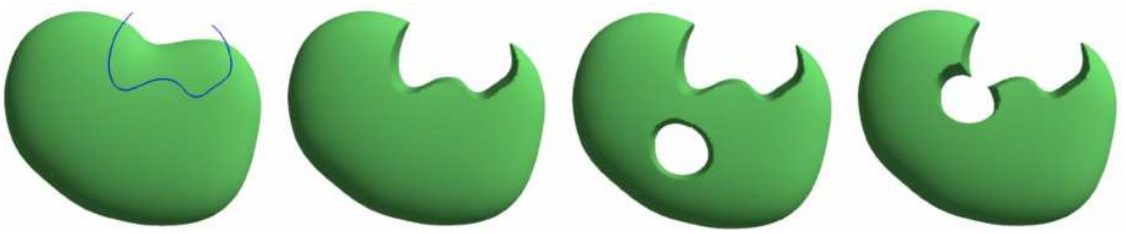

Figure 2.12: Example results for cutting the mesh for creating an aesthetic hole or removing volume from the mesh. Wywill et al. [WFJ+05]

for target shape. Free form deformation can also applicable to sketch-based systems. Draper and Egbert [DE03] proposed a gesture based interface for free form deformation, which allows the user to twist, bend, stretch or squash the model without dealing the technical details of the free form deformation. Severn et al. [SSS06] presented a sketching technique for defining the transformations to models with single stroke. The desired transformations for the model are obtained by using a principle component analysis approach. Sketch-based deformation has been in use for giving poses to characters in animation sequence. Davis et al. [DAC+03] introduced an sketch-based interface for rapidly creating 3D stick figure animation. They first reconstruct the possible 3D configuration and then present the most likely 3D pose for each frame by applying a set of constraints and assumptions. Thorne et al. [TBvdP04] presented another sketching system for character animation, which allows to sketch the motion of the character and after this sketch is mapped to the previously defined motions. Chen et al. [CON05] used the sketching in their animation method, which allows the user sketch the character movements for each animation frame with texture mapping. We will discuss the details of sketching for facial and character animation in the following section.

Selecting the suitable surface representation is a significant factor to determine the quality of the results. Applications are designed according to the selected surface properties because each surface type shows different reactions to the performed operations. For example, parametric surfaces are a known representation and their integration is an easy process during implementations. They are parametrized over 2D coordinates and include spline patches, rotational blending and surface of revolution. This type of surfaces exist in [EHBE97], [CSSJ05]. Mesh based surfaces are widely used representation over years and support arbitrary geometric topologies. Applications developed in [IMT99], [DDGG05], [NSACO05] supported the mesh surfaces. The other common surface type is implicit surfaces, which support blending, boolean operations and hierarchical modeling [AAJ03], [AGB04], [SWSJ06]. Another type

is fair surfaces used in [NISA07], [RSW+07] which is the result of the constrained optimization problem solution where the definition of new constraints causes a re-computation for the result.

The sketch-based interfaces can be designed in either suggestive approach or gestural approach. In suggestive interfaces, the system is implemented with all possible interpretations and user determines the desired interpretation to perform [EHBE97], [IMKT97], [IH01], [TBSR04], [SWSJ06]. In gestural interfaces, an individual stroke determines the operation with a specified sketch recognition method [PJBF00], [DE03], [TBvdP04], [KS06], [OSS07]. The sketching interface proposed in this dissertation has been designed in gestural approach for its practical and simple usage for both experienced and novice users.

2.2.4 Sketching for Facial and Character Animation

The sketch-based interface to manipulate the rigged 3D models is a natural, useful and practical tool for artists and animators. Constructing a sketch-based platform with a significant accuracy gives successful results. Singh et al. [SF98b] presented the wire curves to deform the shape of an object and domain curves to define the domain of the deformation. Pyun et al. [PSS04] formulated wire curves for deformation based on vertices displacements. Nealen et al. [NSACO05] developed a sketching interface, which allows the user to directly sketch on the 3D model incorporating with the Laplacian surface editing techniques for determining the new silhouettes. Nataneli and Faloutsos [NF06] presented a sketching interface for animating blendshape facial models. Zimmermann et al. [ZNA07] proposed an over-sketching interface that finds the nearest silhouette on the model to the sketched line and automatically deforms that particular sub-region of the model by fitting to the line again by using Laplacian deformation components. Chang and Jenkins [CJ08] offered a method, which uses reference and target curves to create articulation and posing of 3D facial models for finding optimum pose within these sketched input curves. Sucontphunt et al. [SMND08] introduced an interactive posing system, which enables manipulation of 3D facial model through control points located on predefined curves on the corresponding 2D portrait. Lau et al. [LCXS09] developed another interactive system for posing 3D facial models that enables the artist to model or edit the face by using free-form strokes with an underlying formulation, which eliminates user given facial poses by comparing with a set of facial expression data. Gunnarsson and Maddock [GM10] proposed a sketching system without any manipulation feature, which allows the artists sketch a

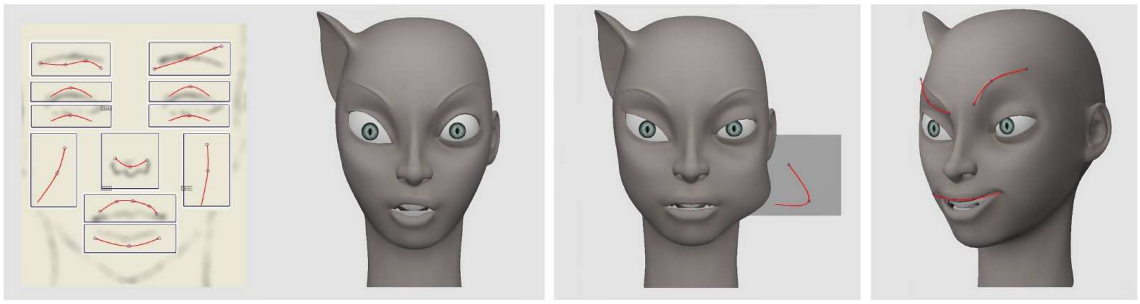

Figure 2.13: The interface and results from Sketching Interface Control System, which allows to sketch both 2D predefined canvas (left) and directly on 3D model (right). Miranda et al. [MAO+12]

desired facial pose and the system builds 3D model by fitting already sketched strokes. Miranda et al. [MAO+11] introduced the external canvas concept, 2D drawing area, which allows the artist rapidly manipulate the facial rig in real-time by drawing simple strokes on the region based external canvas. Once more, same authors [MAO+12] subsequently improved their previous method to allow manipulation of the facial rig directly in 3D space (see figure 2.13).

Oztireli et al. [OBP+13] offered a differential blending algorithm, which works under a sketching interface for curving the bones of whole body rigs to generate poses and animation. Recently, Hahn et al. [HMC+15] proposed a sketched-based system to manipulate whole body rigs for characters animation with an optimized approach, which minimizes nonlinear iterative closest point energy to determine rig components that fit the character's sketch abstraction to an input sketch (see figure 2.14).

Guay et al. [GCR13] developed a mathematical model for line of action, which enables 3D character posing by using single input curves. Moreover, Guay et al. [GRGC15a], [GRGC15b] improved the previous method by adding physically based model and space-time abstraction in the sketching framework.

2.3 Position Based Dynamics Approaches

Almost three decades, physically-based methods have been an active research topic for solid deformations in the field of computer graphics. There exist valuable state of the art reports and courses on this subject such as [WB97], [NMK+06], [MSJT08], [BETC12]. Besides, Bender et al. [BMOT13] presented a detailed state of the art report on position-based methods for solid simulations. Moreover, a notable tutorial from Bender et al. [BMM15] exists on position-based methods.

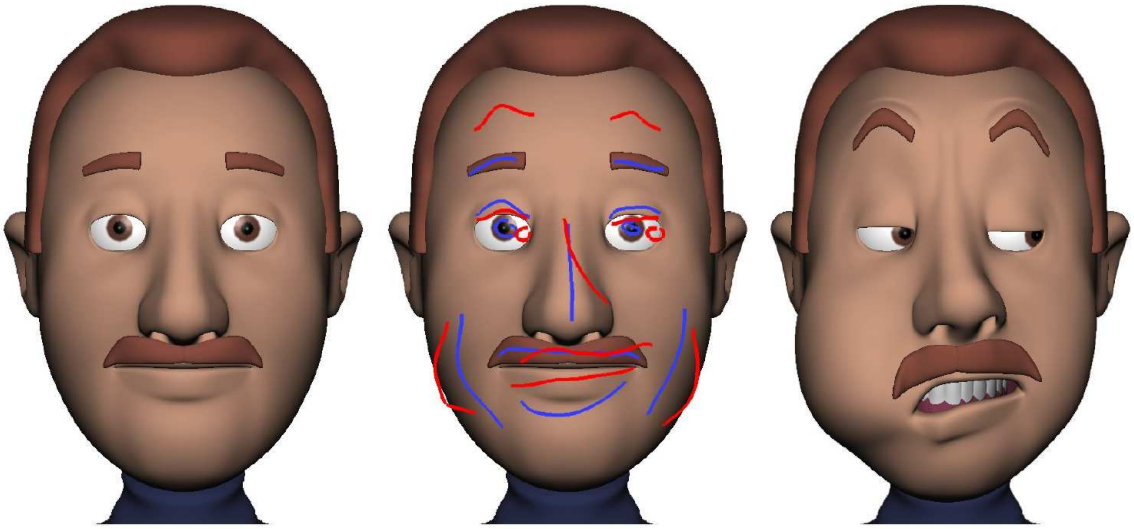

Figure 2.14: Example result of posing the face with multiple stroke pairs. In the stroke pairs, the blue stroke represents the reference curve and red one represents the target curve for the desired manipulation. Hahn et al. [HMC+15]

Foundations of the position based dynamics method are proposed in [Jak01], [MHHR07] and [Sta09]. Jakobsen [Jak01] presented the main idea behind rigid and soft body simulations in Hitman: Codename 47 game engine. He described Verlet integration scheme and direct position manipulation for deforming the rigid and soft bodies in detail based on constraints for interactive gaming. Müller et al. [MHHR07] offered the position based dynamics method, which updates the velocities explicitly instead of an implicit velocity calculation and presented a fully general approach for projected constraints (see figure 2.15). Besides, Stam [Sta09] explained the theory on projection of constraints in Nucleus system similar to position based dynamics method.

The earlier research work on deformable models dating back to the 80's, Terzopoulos

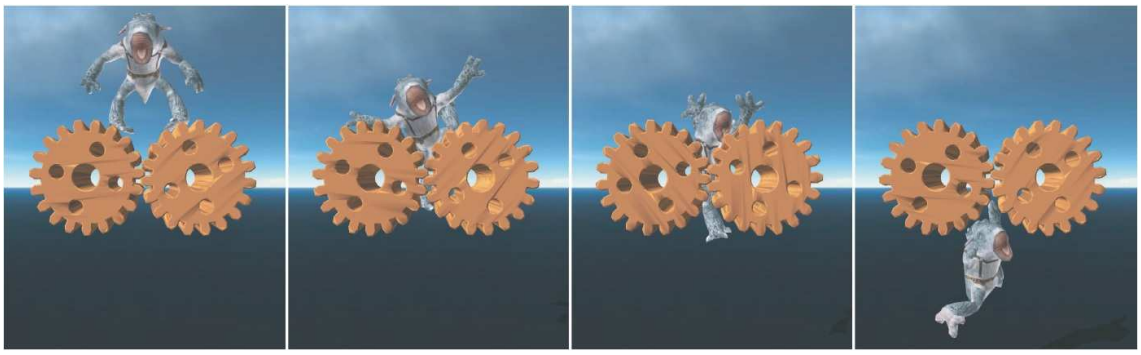

Figure 2.15: Example of a cloth character deformation under pressure by using position based dynamics approach. Müller et al. [MHHR07]

[TPBF87] introduced the simulation of deformable solids by using the elasticity theory. Provot [Pro96] offered to constrain the length of the spring in a mass-spring cloth simulation for preventing the over-stretching of the springs. Baraff and Witkin [BW98] used implicit Euler method and defined general constraints by a function. Faure [Fau98] presented the linearized displacement constraints, which incorporate an iterative solver to find the new positions. James and Pai [JP99] presented an algorithm for real time animation in their system called ArtDefo with a boundary element method (BEM) for discretization. Desbrun et al. [DSB99] used the constraints projection approach with an implicit integration method to handle stable and robust mass-spring system for virtual reality.

In following years, the research in the field was accelerated, Bridson et al. [BMF03] used projection of constraints with an implicit/explicit time integration scheme and a novel contact collision method. Grinspun et al. [GHDS03] used the constraints for the simulation of a variety of curved objects ranging from paper to metal. Irving et al. [ITF04] presented an algorithm for large deformations on elasto-plastic solids by using a finite element simulation approach. Müller and Gross [MG04] offered a fast approach in real time applications for simulating elasto-plastic materials and fracture by combining the extended stiffness finite element approach with strain based plasticity model. Teschner et al. [THMG04] demonstrated a versatile and robust model for elastic and plastic deformations of the geometrically complex objects, which can handle a significant variety of materialistic properties from rigid to fluid state. Choi and Ko [CK05b] proposed a method, which identifies the rotations from the small deformations and extend the base modal analysis formula to accommodate the identified rotations. Teschner et al. [TKZ+05] presented a notable state of the art report on collision detection for deformable objects. Müller et al. [MHTG05] offered the shape matching concept for elastic and plastic deformations in a meshless approach, which does not need any connectivity information such as point-based models (see figure 2.16).

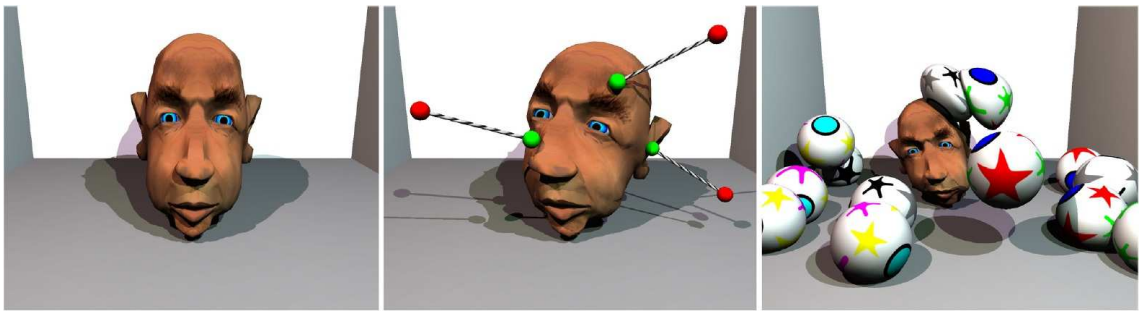

Figure 2.16: Example test scene of shape matching to demonstrate user interaction on a deformable head model and collision handling with other deformable objects. Müller et al. [MHTG05]

Hong et al. [HJCW06] proposed the fast volume preservation in a mass-spring system for realistic deformation effects in animations with a low computational cost. Funck et al. [vFTS06] defined shape deformations by divergence free vector fields for realistic looking results with volume conservation and deformations do not include self-intersections. Wang et al. [WXX+06] proposed a mass-spring model for virtual reality systems based on shape matching, which connects the surface with a rigid core using the spring forces. Irving et al. [ISF07] presented a numerical method, which conserves the volume locally in a finite element mesh.

Figure 2.17: Hierarchical position based dynamics approach offers a multi grid solver, which enables high resolution cloth simulation in real time. Müller [M.08]

The geometric constraint projections gave a notable impact to the research, such as Bergou et al. [BMWG07] presented the Tracking process, which uses the constraints to enhance the surface deformations with physically simulated details. Goldenthal et al. [GHF+07] offered a novel fast constraint projection method to gain low strain in warp and weft directions. Kubiak et al. [KPGF07] offered a real time thread simulation, which computes the stretching, bending, torsion and provides output forces for haptic feedback. Müller [M.08] proposed to speed up the traditional position based dynamics method with a multi-grid based approach by keeping the origins of constraint projections (see figure 2.17). Steinemann et al. [SOG08] presented a method for shape matching deformation model on many overlapping regions to handle topological changes by using an octree-based hierarchical sampling. Stumpp et al. [SSBT08] proposed an adaptive shape matching approach for cloth simulation, which uses two different type of clusters for high stretching, shearing and low bending resistance. Schmedding et al. [SGT08] improved the damping approach of iterative spring damping concept for stability improvement, which can be used on any arbitrary

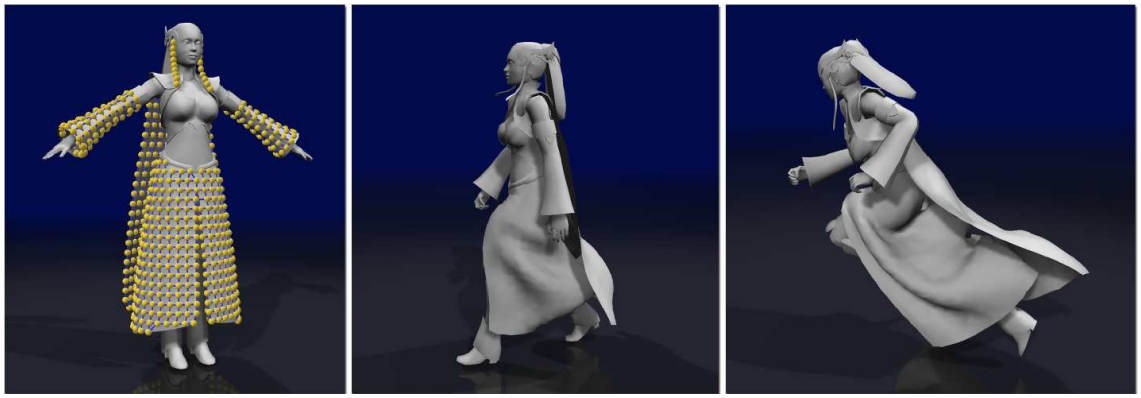

Figure 2.18: Example simulation test scene for the character animation with additional physics and oriented particles. Müller and Chentanez [MC11]

position-based deformation models.

In recent years, various techniques have focused on the quality and efficiency, for example, Kelager et al. [KNE10] offered a novel bending approach for high quality and faster simulation, which is an alternative to dihedral bending method. Müller and Chentanez [MC10] presented the wrinkle meshes for important details in cloth or skin surface of a character by attaching a higher resolution wrinkle mesh to the coarse mesh. Wang et al. [WOR10] described a strain-limiting approach, which acts on strain tensors in a constant coordinate form, which allows fast efficient simulation. Diziol et al. [DBB11] introduced GPU based real-time approach to simulate incompressible objects with huge number of elements by using a shape matching method, which includes an extended version fast summation technique for triangulated meshes. Müller and Chentanez [MC11] proposed a method for rapidly simulate different types of objects by using oriented particles, which include the rotation and spin as an additional information for traditional particle approaches (see figure 2.18).

Kim et al. [KCMF12] introduced the long range attachment approach, which attaches the character cloth to the kinematics parts of the character to prevent the inappropriate stretching. Zurdo et al. [ZBO13] offered a fast example-based algorithm for cloth wrinkles, which defines wrinkles as a function of low-resolution cloth and example poses set. Bender et al. [BWD13] proposed a method for the deformation of complex stiff cloth models by using a novel shape matching method and only focus to stretch and shear stiffness. Deul and Bender [DB13] introduced the multi-layer model for physically-based skinning application by using oriented particles concept incorporating with the position-based constraint projection. Bender et al. [BKCW14] offered a simulation method based on continuum mechanics for solid deformation, which focuses

Figure 2.19: Continuous materials simulated with a position-based approach. The visual results obtained from the simulation are consistent with behavior of real world materials. Bender et al. [BKCW14]

on physical effects such as elasto-plasticity, anisotropy or lateral contraction (see figure 2.19). Deul et al. [DCB14] presented a position-based method for rigid bodies by solving the position constraints between rigid bodies. Müller et al. [MCKM14] introduced the new constraints to position based dynamics method, which all the control of the strain directions independent from the edge directions. Fratarcangeli and Pellacini [FP15] parallelize the position based dynamics method by using a practical partitioning technique with a graph coloring algorithm.

2.4 Summary and Discussion

In this chapter we have summarized the popular and relevant works, which have been presented by the authors in the area of facial animation, sketching interaction and dynamically deforming solids. Despite taking the advantage of the all work mentioned in this chapter, the core of our research is inspired by Lewis and Anjyo's *blendshape direct manipulation* [LA10], Miranda et al.'s *sketch express* [MAO+12], Jakobsen's *advanced character physics* [Jak01], Müller et al.'s *position based dynamics* [MHHR07] and Stam's *Nucleus* [Sta09].

In the scope of this dissertation, we have proposed a set of solutions and additional features for the open issues, which have been required further development. For example, creating blendshape facial animations using traditional weight editing tools requires either memorizing the function of a large number of parameters, or a trial-

and-error search in a high-dimensional space. Direct manipulation technique [LA10] addresses this problem, allowing the artist to directly pin and move manipulators placed on the surface of the face. However, this technique gives an unintuitive global deformation impact to the face. In our solution, we propose a novel method, which overcomes this problem and keeps the deformation in local basis as intended. Besides, we introduce a sketch-based interface as an application to our solution inspired by sketch express [MAO+12]. Although sketch express was focused to manipulate the facial articulated joints, our sketching interface manipulates the desired blendshapes. Nevertheless, we have also followed continuous manipulation principle of the sketch express. This principle means generating complex deformations with only one continuous sketched curve. Besides our *sketched based direct manipulation* method, we introduce another novel method called *transposition based direct manipulation* which addresses the instability of the pseudo-inverse based mathematical formulations of the blendshape direct manipulation techniques.

Furthermore, position based approaches vary according to its geometric constraints and time integration scheme. [Jak01] used verlet integration by calculating the velocities implicitly. [MHHR07], nevertheless, calculated the velocity explicitly and the following research sustained explicit velocity calculation. In our research, we have developed a new framework, which is focused on simulation of the deformable surfaces with verlet integration by combining *geodesic circles* and *position-based dynamics* fashion projected geometric constraints. However, the existing research applications deform the models as a whole. Our work allows an interactive selection of the deformation influence area over the model by computing the geodesic distances from an arbitrary point to a desired range. This selection keeps the surface deformation inside the boundaries of the area and outside of the boundaries the model conserves its rigid behavior. Our solution allows applying the position-based dynamics method to highly complex objects such as face efficiently.

To sum up, the main research goal of this dissertation is to offer practical solutions to create complex deformations on facial models to obtain high quality results with a simple and intuitive interaction model.

Chapter 3

Direct Manipulation of Blendshapes Using a Sketch-Based Interface

We introduce a method that localizes the direct manipulation of blendshape models for facial animation with a customized sketch-based interface. Direct manipulation methods address the problem of cumbersome weight editing process using the traditional tools with a practical "pin-and-drag" operation directly on the 3D facial models. However, most of the direct manipulation methods have a global deformation impact, which give unintuitive and unexpected results to the artists. To this end, we propose a new way to provide local control the theory of direct manipulation method by using geodesic circles for confining the edits to the local geometry. Inspired by artists' brush painting on canvas, we introduce a sketch-based interface as an application that supports direct manipulation and produces expressive facial poses efficiently and intuitively. Our method allows the artists to simply sketch directly onto the 3D facial model and automatically produces the expeditious manipulation until the desired facial pose is obtained. We show that localized blendshape direct manipulation has the potential to reduce the time-consuming blendshape editing process to an easy freehand stroke drawing. Finally, we present a wide range of facial poses on various models that are created rapidly using our method.

3.1 Motivation

Facial animation in movies and video games remains a challenging problem for the area of computer graphics. The major reason is the complexity of authoring the rig underneath the face to create convincing and meaningful facial expressions. Blendshape is a widely adopted technique for high-quality facial rigging. Conceptually, the technique of blendshape is the simple linear model of facial expression. Despite its simplicity, producing facial animation using blendshapes requires a serious effort and time to achieve realistic high quality facial models that includes hundreds of blendshape targets.

Blendshape model forms a specific facial expression as a sum of linear combination of target faces that are obtained either by scanning a human face with various expressions or by a skilled artist sculpts each expression manually. Once all expressions are assembled, facial poses can be expressed with an insignificant computational effort by altering the weights of the linear combination. Traditionally, these weights have been edited directly by the artists, using an interface in which each weight is presented as a slider. However, creating blendshape facial animation using traditional weight editing tools requires either memorizing the function of a large number of parameters, or a trial-and-error search in a high-dimensional space. Direct manipulation blendshapes [LA10] addressed this problem using a mathematical framework that allows the artist to perform "pin and drag" operations on the surface of the 3D model. *Pinning* is the selection of an arbitrary point from the model surface, and *Drag* is the movement of the pinned point on the surface of the model. During this *pin and drag* operation, the underlying method solves an inverse problem of finding the corresponding weights. The mathematical framework of direct manipulation aims to change the shape of an arbitrary area of the face. However, it may cause slight possible alterations in the remainder of the face. During the drag, if the point movements are applied in a short range, the change in the remainder of the face is negligible. However, if the artist applies a long range drag, the resulting facial expression is unpredictable, and can even reach a unstable and noisy state. This peculiarity limits the freedom of the artists during the process of preparing facial poses for keyframe animation.

In this chapter, we propose a new method which optimizes the mathematical framework of blendshape direct manipulation by localizing the desired facial deformation during the artistic pose editing. During the pin and drag operation of the direct manipulation technique, we observe that altering the blendshape weights through the pin drag gives a slight deformation impact to almost all vertices of the model. For

example, while a pin movement attempts to raise one of the eyebrows, the mouth or jaw slightly moves unexpectedly. The proposed method focuses on confining the artistic edits to the local facial geometry by using geodesic circles. We rebuild the predefined deformation elements (here, blendshape targets) to overcome this unexpected global deformation impact which arises from the mathematical framework of direct manipulation. This operation is performed by computing the *heat kernel* [CWW13] to create geodesic circles around each constraint point and limit the edits according to the existing targets within the circle. The method analyzes all target shapes over the selected vertex and modifies the input data according to the position of the vertex. This operation is done automatically without any manual intervention.

We also present a sketch-based interface which incorporates our localized direct manipulation. Recently, the tendency for sketch-based interfaces has increased as a way of creating animation sequences, deforming shapes or controlling objects. With sketch-based interfaces, artists work with natural movements, analogous to charcoal drawings, instead of having to deal with hundreds of different rigging and deformation control parameters using the external interfaces. Most of the facial sketching methods (such as [MAO+11] or [MAO+12]) work on bone-based rigs and allow the direct manipulation with sketched curves. However, in our case, the model is rigged with blendshapes which include more details for each target pose and excessive parametric combinations to gain semantically meaningful poses. In our sketching procedure, the user can draw a simple freehand stroke onto the 3D facial model at any angle and point of view independently. After, the method transforms this simple stroke into a "pilot" curve automatically. Pilot curve not only determines the positions of the manipulators to the desired locations on the 3D model but also creates a blendshape deformation to the model. We mimic the drag operation of direct manipulation under the *horizontal curves* concept [Lev07] by using curve's own entities such as curvature depth, radius, angle, etc. This sketching process can be repeated until the desired facial expression is obtained for each frame of the animation sequence.

We present our method's flexibility and easy usage over different models demonstrating the creation of several facial poses in Autodesk Maya [Aut]. Our method can be used on any blendshape facial model without any preceding operation. The whole pipeline is automated just to keep the user away from the technical details of the method. Even though in this work we target the experienced users, the novice users can adapt to our system without any difficulty and easily learn the details of the facial rig concept by using our approach. As a result, our work offers a fast, powerful and easy to implement approach for blendshape animation by creating a bridge between 2D curves and 3D

model.

3.2 Method

The proposed method operates under a sketch-based interface, which has been created in Maya. Our approach includes 4 discrete steps: first, a freehand stroke is applied the on the imaginary canvas, which is a 2D drawing region aligned to the image plane. This stroke is converted to a NURBS curve, and mapped on the 3D facial model. Second, the knots of NURBS curve are intersected with the closest points that belong to the model mesh. Thereby, the manipulators (*pins*) are constructed on these points. Third, fuzzy support fields with geodesic circles are created around each pin for localization. Finally, the blendshapes of the model are manipulated by using the curve's entities. This whole process can be repeated until the desired expression is realized. Figure 3.1 demonstrates a brief overview of our method.

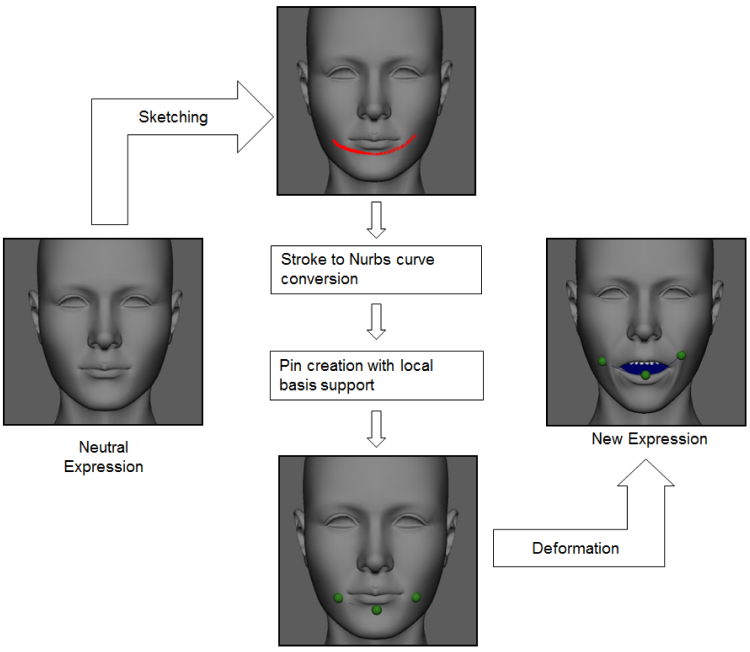

Figure 3.1: A Brief Method Overview.

3.2.1 Sketching Representation

The free-form stroke (S) can be defined as a set of ordered points $\{s_0, ..., s_{n-1}\}$, which represents the stroke in screen space. Drawing is originally applied on a dynamically

created virtual canvas that is oriented orthogonal to the view direction.The direction of S is determined by the user during drawing. S is projected from the virtual canvas onto the closest visible surface of the 3D model by a ray tracing operation (see Figure 3.2).After the projection, a Nurbs curve (N) is generated on the surface of the model with the same length of the projected stroke. N plays a critical role in our method, because it determines the exact positions of the pins.Therefore, we restrict its shape to a spline space for simplicity, which protects the user from the undesired details due to the processing load of the computer during recording of the mouse movements. A cubic spline is selected as the best choice, because its simplicity and easy to obtain a close approximation to the surface of the 3D facial model. A lower degree curve would not lie on the surface, while a higher degree curve may incorporate unintended details of the artist's gesture. Although the cubic spline generally does not lie exactly on the surface of the model, this approximation is not obvious or obtrusive for the artist.

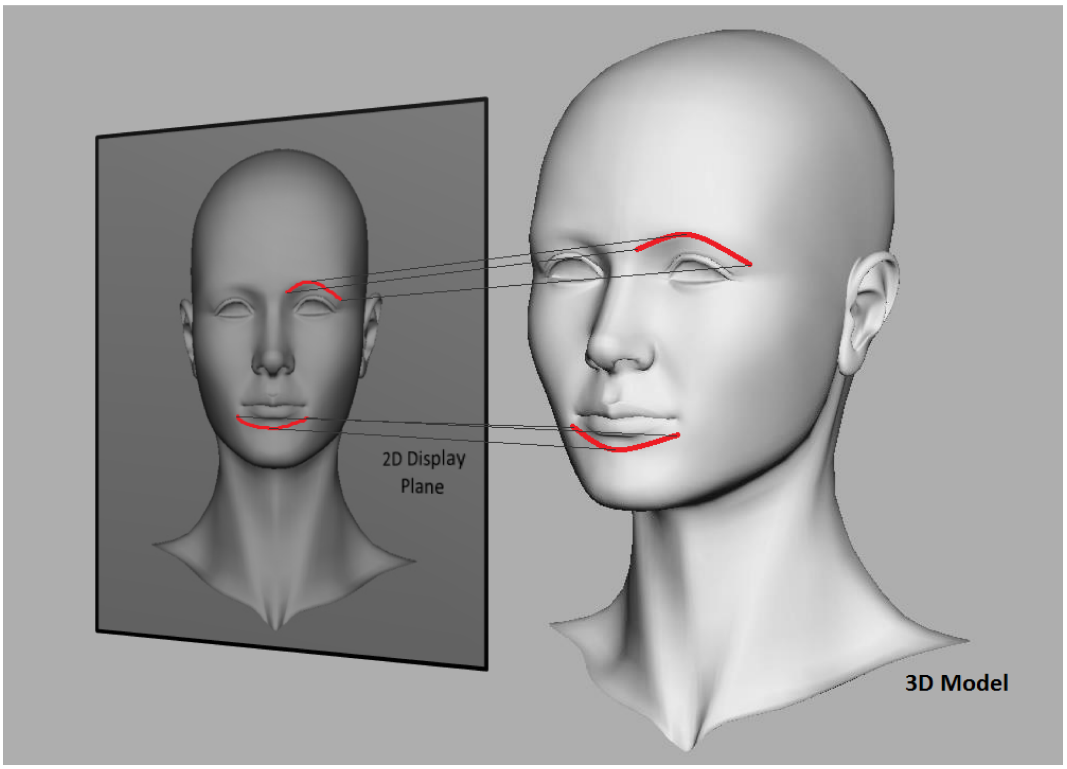

Figure 3.2: Application of the mapping of free-form stroke from 2D display plane onto the visible surface of the 3D model.

3.2.2 Curve Knots and Mesh Vertices Intersection

We create a physical connection between the Nurbs curve N and the 3D model in the scene, where these objects are geometrically different and independent from each other. After the *Sketching Representation* phase, N with the degree $D = 3$ is located onto the surface of the model and parametrized with k knots, where k corresponds to D. The method creates N by selecting the knots over the total number of points of the original S. These knots have two main attributes: they are homogeneous and located at the beginning, middle and end points of N. To create a direct correlation between N and the 3D model, we have implemented algorithm 1, which intersects each knot with the closest vertex of the 3D model by using Maya's optimized octree method. The most significant consideration in this algorithm should be noted that it avoids to give any unintended noise to the model by changing any component's attribute during the process. Figure 3.3 illustrates the visual result of this direct correlation between N and the 3D model.

```
foreach knot in N do
    intersectPoint ← 0;
    Intersector ← MeshIntersector();
    matrix ← TransformationMatrix();
    objectNode ← DirectedAcyclicGraphNode();
    stat ← Intersector.create(objectNode, matrix);
    if stat ≠ 0 then
        stat ← getClosestPoint(knot_r, pointInfo); k ← pointInfo.FaceID();
    end
    foreach vertex in Model do
        vertexID ← Model.VertexID();
        faceID ← Model.FaceID();
        if faceID = k then
            intersectPoint ← Model.VertexPosition(); knot_r ← intersectPoint;
        end
        else continue ;
    end
end
```

Algorithm 1: Points Intersection Algorithm

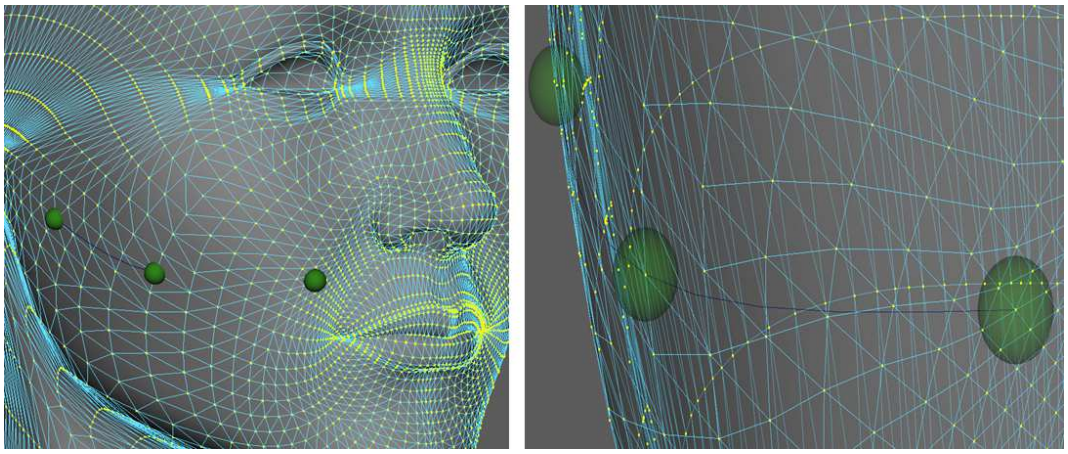

Figure 3.3: Representation of our intersection algorithm. (left) A general view of 3D model with pins and curve. (right) A close X-ray view for the details about the curve, pins and mesh vertices. The algorithm does not violate the originality of the model by altering the position of any component of the 3D model.

3.2.3 Blendshape Direct Manipulation and Localized Deformation

Manipulators (pins), which are represented as small sphere units, are created on the locations of intersected points over the model automatically by our method. In blendshape direct manipulation, the pin behaves as the inverse kinematics (IK) end-effector for determining the values of the position parameters. However, there exists a significant difference between blendshapes and IK handles regarding the parametrization. For example, moving a human arm by using inverse kinematics roughly requires 7 degrees of freedom for the angles of the joints [TB96] but in a realistic blendshape model, the relevant weights can be more than 100. Therefore, we have to explore a discrete function to solve the inverse problem for direct manipulation. The solution of the defined function determines the final weights with the constraints, given by "pin and drag" operation.

According to algebraic perspective, blendshape is a simple vector sum as illustrated in equation (3.1):

$$f = \sum_{k} b_k w_k \tag{3.1}$$

In order to make equation (3.1) simpler, it can be denoted as f=Bw. f is the produced final face and also a vector which contains all vertex coordinates of the mesh in the

order of *xyzxyz...*, B is a matrix whose column vectors, b_k, include the coordinates of individual blendshape targets, w consists of the blendshape weights. However, one face model, f_0, is special because all individual blendshape targets are offsets from f_0, which is called the neutral face shape. It has to be added to equation (3.1): $f = Bw + f_0$ (see equation (3.2) for details). This formulation is called as delta blendshape formulation and is widely used in commercial modeling and animation packages, and in this chapter we use this equation as a base.

$$\begin{pmatrix} f_{1x} \\ f_{1y} \\ f_{1z} \\ f_{2x} \\ f_{2y} \\ f_{2z} \\ \vdots \\ f_{nz} \end{pmatrix} = \begin{pmatrix} x & x & \cdots & x \\ y & y & \cdots & y \\ z & z & \cdots & z \\ x & x & \cdots & x \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{b_1} & \mathbf{b_2} & \cdots & \mathbf{b_n} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} + \begin{pmatrix} f_{0_{1x}} \\ f_{0_{1y}} \\ f_{0_{1z}} \\ f_{0_{2x}} \\ f_{0_{2y}} \\ f_{0_{2z}} \\ \vdots \\ f_{0_{nz}} \end{pmatrix} \tag{3.2}$$

By recalling equation (3.2) for reviewing the background [ATL12], we assume a facial pose is created by using direct manipulation. Pins, thereby, are located on the mesh and dragged by the user. The resultant face can be called as $f = Bw + f_0$. After applying pinning to the arbitrary vertices on the face model, some of them are fixed, some are dragged and the rest of the vertices remain the same. Then, we can call the resultant face also $f = m + f_0$. Here, m denotes for the moved pins. After all, we determine the final weights of the resultant face model by minimizing the residual between Bw and m: $\|\bar{B}w - m\|$ where $\bar{B}$ represents the corresponding rows from the B matrix in equation (3.2). We use the least squares with a regularization term to solve this direct manipulation problem for finding the final weights (see equation (3.3)).

$$\min_w \|\bar{B}w - m\|^2 + \alpha\|w\|^2 \tag{3.3}$$

where α is a user defined regularization parameter which has to be selected carefully for the most optimal fit. Equation (3.3) is a soft constraint linear system which is easy to solve and implement. The final weights of the blendshape model are obtained by taking the gradient of equation (3.3) with respect to w and set it to zero. If we denote equation (3.3) as $E(w)$, then we seek to find $\nabla_w E = 0$ (see equation (3.4)).

$$\nabla_w E = 2\bar{B}^T\bar{B}w - 2\bar{B}^T m + 2\alpha w = 0 \tag{3.4}$$

We reorganize the equation (3.4) to obtain "**Ax=B**" form linear system for computing the blendshape weights (see equation (3.5)) which is the core of our approach.

$$w = (\bar{B}^T\bar{B} + \alpha I)^{-1}\bar{B}^T m \tag{3.5}$$

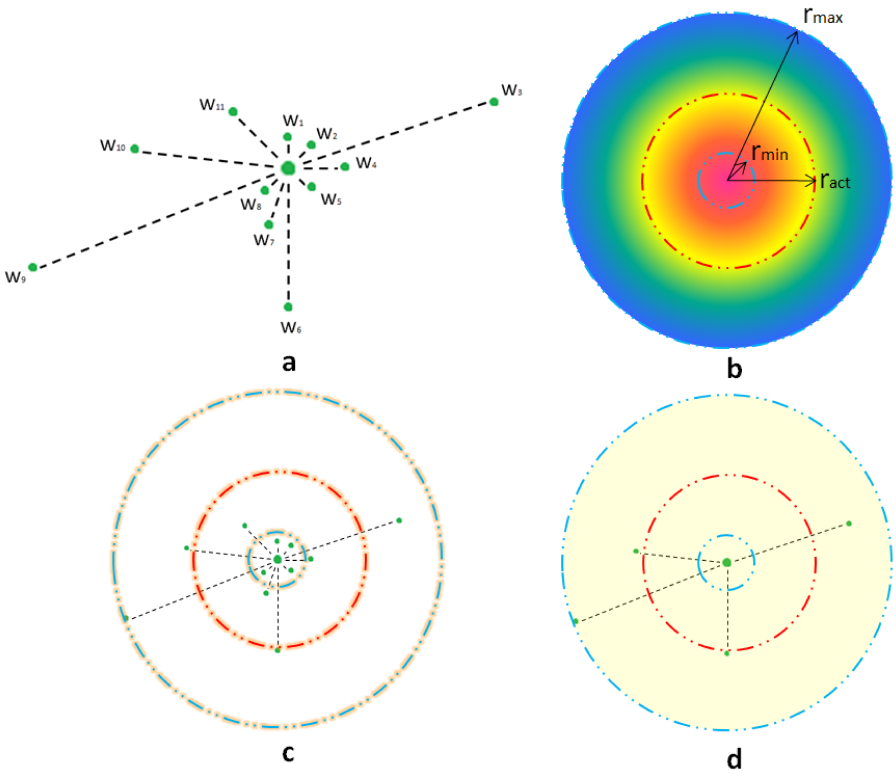

Figure 3.4: (**a**) shows the pinned vertex behavior under the influence of all individual weights. (**b**) shows the invisible support field obtained from the heat method. The ranges are assigned to cover the closest neighbors of the selected vertex with a custom range unit, therefore in our test models we set $r_{min} = 0.1$ and $r_{max} = 0.5$. (**c**) shows vertex behaviors in the support field. (**d**) shows how we eliminate the undesired weights.

The major limitation of direct manipulation is the fact that editing a certain part of the model causes minor changes on the other parts of the face. While the weights are computed during the pin movements, equation (3.5) produces non-zero weights for all target shapes. For example, moving a pin in the right corner of the mouth can cause a small deformation in the left eye-brow. However, if the drag is performed with slight movements, this "side effect" is not that obvious. Nevertheless, if the artist prefers to

work with gross gestures, then the resultant pose can be unintuitive. Briefly, this is because of the almost non-zero content of the $\bar{B}^T\bar{B}$ in equation 3.5. In direct manipulation, weights have a pseudo-inverse relationship with the dragged pins, and columns of the pseudo-inverse tend to have non-zero entities. Even though the regularization term in equation (3.3) can be improved by using different parametrization techniques, the resultant method may not produce meaningful poses because of this instability. Therefore, we iteratively analyze each column of $\bar{B}$ from equation 3.2 to observe the deformation impact of each blendshape target. Without any exception, each column of $\bar{B}$ produce a deformation impact on the pinned vertex. Some targets have a notable deformation effect on the pinned vertex and some have very little (almost negligible) according to the influence area of the target shapes. As a consequence, we create a fuzzy support field around these pinned vertices in run time by using the heat method from [CWW13]. The purpose is to eliminate the targets, which give the least deformation to the area of the pinned vertices (see figure 3.4).

The support field is created by computing the geodesic circles from the pinned vertex to its neighbor vertices (see figure 3.5). The pinned vertex stays in the center of the circular support field. We calculate the geodesic distance for the radius of the fields as short as possible without losing all the deformation on the pinned vertex. We define the size of this field in a range of $[r_{min}, r_{max}]$ shown in figure 3.4. r_{min} is the radius of the inner circle, and r_{max} is the the radius of the outer circle. Further, we set an intermediate radius that is the actual radius r_{act}, as a certain limit to eliminate weights which have the least deformation impact on the pinned vertices. For the models shown in this chapter, r_{act} is calculated as the linear average of the distances r_{min} and r_{max} ($r_{act} = \frac{r_{min}+r_{max}}{2}$). The conditional relationship between $\bar{B}$ and r_{act} is created iteratively during the pin generation (equation (3.6)).

$$b_i{}^k = \begin{cases} 0, & \text{if } |b_i{}^k| < r_{act} \\ (b_{ix}, b_{iy}, b_{iz})^k, & \text{if } |b_i{}^k| \geq r_{act} \end{cases} \tag{3.6}$$

where b_i is the column vector of $\bar{B}$ that is ($\bar{B} = [b_1, b_2, ..., b_n]$) with n blendshape targets. Each triplet ($b_i = (b_{ix}, b_{iy}, b_{iz})^k$) originally corresponds to the x, y, and z displacements of the pinned vertex k under the influence of blendshape target i. In summary, the range of the geodesic circles determines the desired deformation space for the arbitrary part of the facial model. Our method takes r_{min} and r_{max} as tunable user-specified parameters for the range of the geodesic circles. Users of our method, thereby, have to determine the most optimal values according to the facial

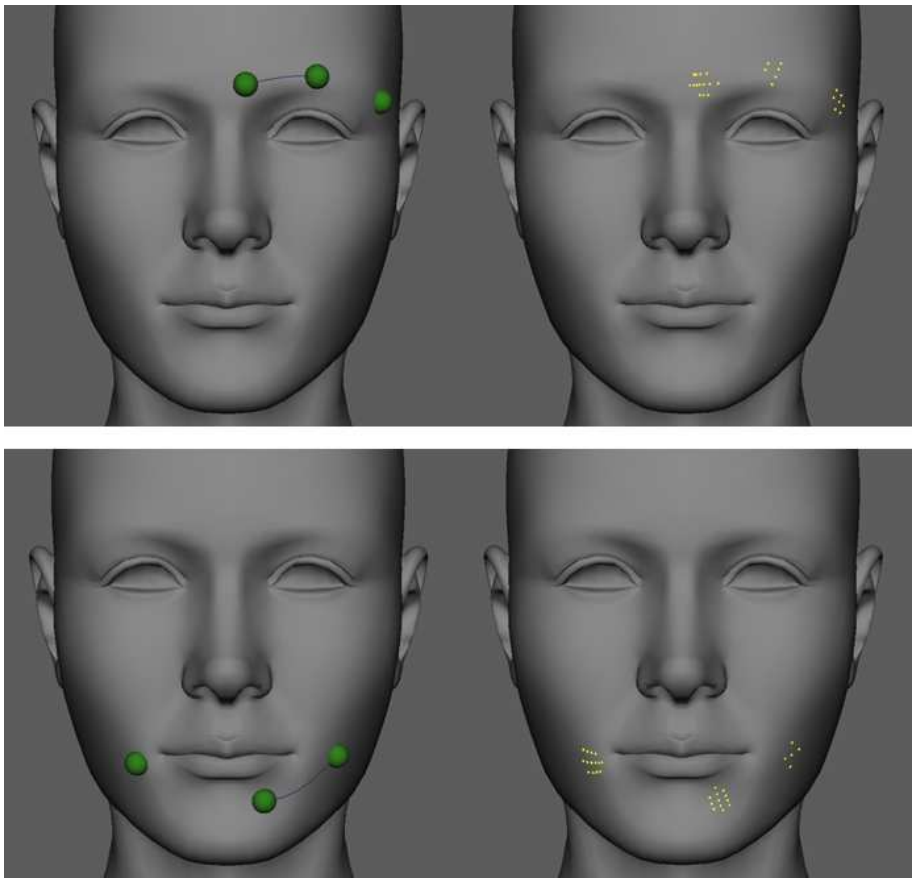

Figure 3.5: Effect of the heat method illustrated on the blendshape face model. In both images, the heat method provides the boundaries of the inner and outer circles (r_{min}, r_{max}) to obtain the closest neighbor vertices of the pinned vertices (the ones in center of each yellow point group) by calculating the geodesic distances. Fuzzy support regions cover these neighbor vertices for localization.

model. For example, in our tests we observe two conditions: If we keep $r_{act} = r_{min}$, some weights, which have the least influence on the pinned vertex, remains in $\bar{B}$ and causes unexpected movements during the pin drag. On the other hand, if we keep $r_{act} = r_{max}$, the essential weights for the pinned vertex can be excluded from $\bar{B}$ and the pin movements may not give the desired deformations. These boundaries can be re-parametrized easily according to the size and blendshape target density of the model.

The optimized $\bar{B}$ serves as the input matrix in equation (3.5) for determining the final weights of the blendshape model. This process confines the weights within the boundaries of the defined geodesic circles, and therefore keeps the deformations in local facial geometry without any global impact during the artistic pose editing with pin movements. For the details of the heat method, we refer the reader to [CWW13].

3.2.4 Deformation with Curve Entities

Drag is the movement of a pin from its rest position to any desired location on the surface of the 3D model. However, we offer this drag operation by using the curve entities which are already sketched on the model. The major challenge of using one single curve to create the deformation is the unknown parameters for the initial and final positions. Either direct manipulation or sketch-based deformation methods require the definition of initial and final position parameters from the user or the system. To overcome this problem, we propose a fundamental estimation to accept the curve as a bended wire from its straight state (see figure 3.6). This estimation defines a movement path, which mimics the pin drag with its initial and final positions.

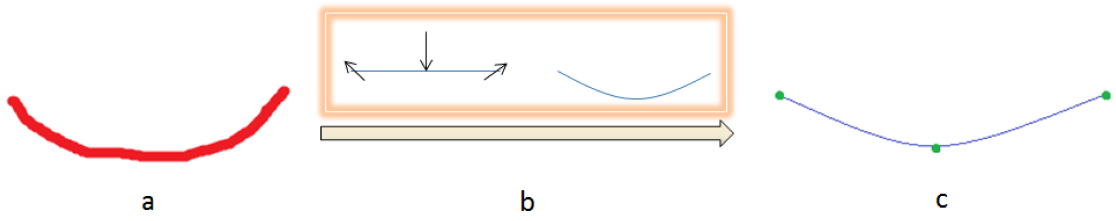

Figure 3.6: Transition from the stroke to Nurbs curve with the bended wire estimation. (a) shows the applied stroke as an input. (b) shows bended wire estimation. (c) shows the final state of the Nurbs curve with pins.

A good example for our approach is the *horizontal curves* concept [Lev07] which are widely employed in the field of transportation engineering for curving the highways with a smooth transition. The horizontal curves are, by definition, circular curves with a certain radius. We take the advantage of the key elements of horizontal curves for our calculations such as radius (r), middle ordinate (M), and central angle (θ) (Figure 3.7). These elements provide an intuitive deformation transfer between the projected spline and the pins. The initial positions of the pins are the positions of the actual intersected vertices on the model, which is explained in section 3.2. Our main goal is to obtain the middle ordinate (M) of the curve and the stretching of the side pins (figure 3.6b and 3.7) for finding an approximated deformation displacement path. Moreover, we employ Maya's own tools to obtain the radius of the curve. With the curve angle (θ) and radius (r), we calculate M with an easy formula, $M = r \times (1 - \cos(\frac{\theta}{2}))$. M is only used for the middle pin, nevertheless there exist two more pins on the same curve. For increasing the visual impact, we stretch the side pins using the value of M. We allow both pins to move in the range of $M \times \cos(\frac{\theta}{2}))$ for x and $M \times \sin(\frac{\theta}{2}))$ for y directions (see figure 3.7). The spline curves frequently do not align on a perfect form

of horizontal curve as shown in figure 3.7, however they have the tendency to keep the shape as close as possible to the horizontal curve. After the deformation obtained by the curve, the user can fine tune the manipulation by selecting one of the pins on the curve and drag it until the desired deformation is obtained.

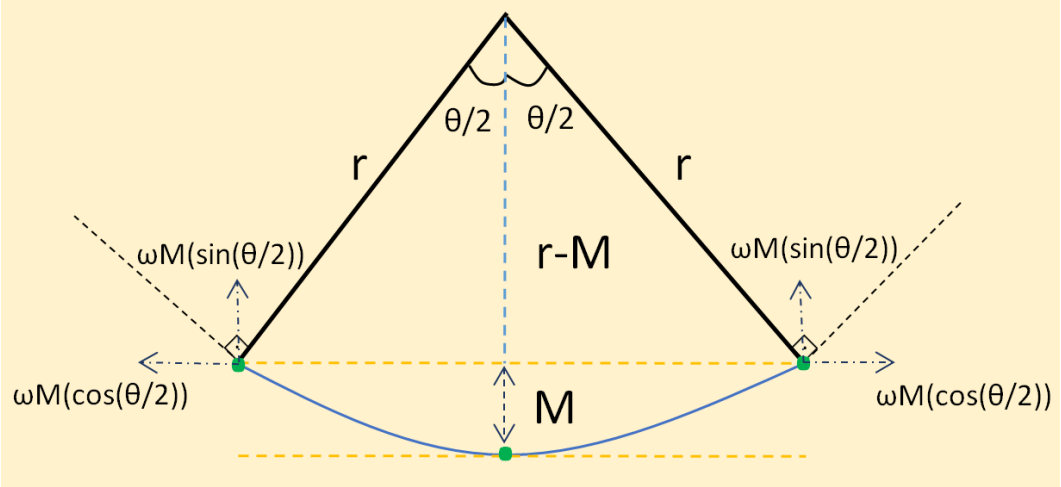

Figure 3.7: Simple demonstration of the Horizontal Curve approach for deformation with curve entities.

Model size is a major dependency regarding the sketched curve manipulation. Curve length, curve radius, curvature size and middle ordinate inherently change according to the size of model. Therefore, curve entities should be adjusted for the most optimal fit to the model size. Although the curve size is determined by the user during the stroke drawing, in some cases the curvature depth may not be enough for the expected result on a large size model or the stretching of the side pin may result wider than expected for a small size model. For that reason, we assign user-specified scaling multipliers (β, ω) to middle ordinate (M) and stretching of the side pins that is illustrated in figure 3.7. These multipliers can be determined or modified easily by the user according to the model size.

The deformation direction is another issue to determine. Although the curve lies on the surface of the model and calculates the displacements, the interface needs the proper manipulation direction collinear with M. Therefore, after each time the curve is generated on the model, we explore its curvature direction from the middle point. If the normal in y direction of the middle point of curve is negative, the concavity is downward. On the other hand, if the normal in y direction of the middle point of curve is positive, the concavity is upward (see figure 3.8).

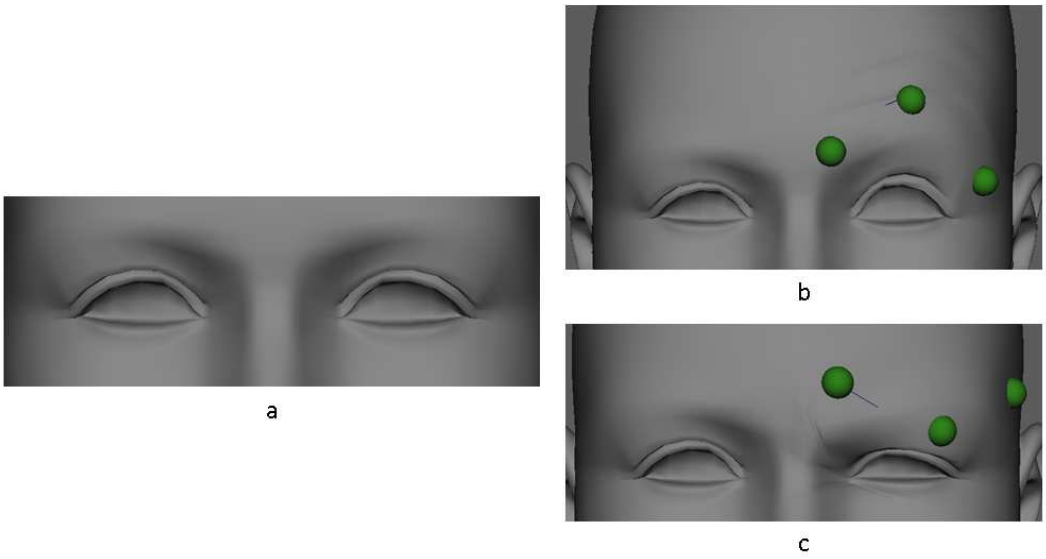

Figure 3.8: (a) shows the neutral pose of the eyes. (b) shows the curve applied with downward concavity. (c) shows the curve applied with upward concavity.

3.3 Validation and Experimental Results

Our method and its interface have been implemented as a plugin for Maya 2016 [Aut] using C++ and Python. Numpy, Scipy and Scikits.sparse packages have been used for matrix calculations. In our method, all steps respond in real time and rapidly produce the interactive visual feedbacks to the user, so there is not any involvement of a specific operation before or after the execution. Our main motivation has been keeping the user away from the technical aspects of the method during the implementation. For that reason except the mouse, the interface does not include any control component which may create some confusion to the user. All the tests and validations were performed on 4-core Intel Core i7-2600 3.4 GHz machine with 8 GB of RAM and an nVidia GTX 570 GPU.

3.3.1 Regularization Parameter

Although our major focus has been the input matrix $\bar{B}$ in the proposed method, α in the regularization term of equation (3.3) plays a critical role in our approach. Most importantly, α keeps the equilibrium by fitting the new weight values according to the desired pose during the weight update. If the value of α is set to zero ($\alpha = 0$), the method shifts to an unsteady state and inevitable haphazard shapes occur. On the

other hand, setting α to higher values ($\alpha > 1$), then the regularization term completely dominates the equation (3.3) and the user never gets any respond from the system. The selection of alpha parameter is done according to the most optimal fit between the pin movement and corresponding deformation. We set the value of α to a small number ($\alpha = 0.0001$) according to the meaningful visual responses from the models. The deformation results with different α values for the same pin movements are shown in figure 3.9.

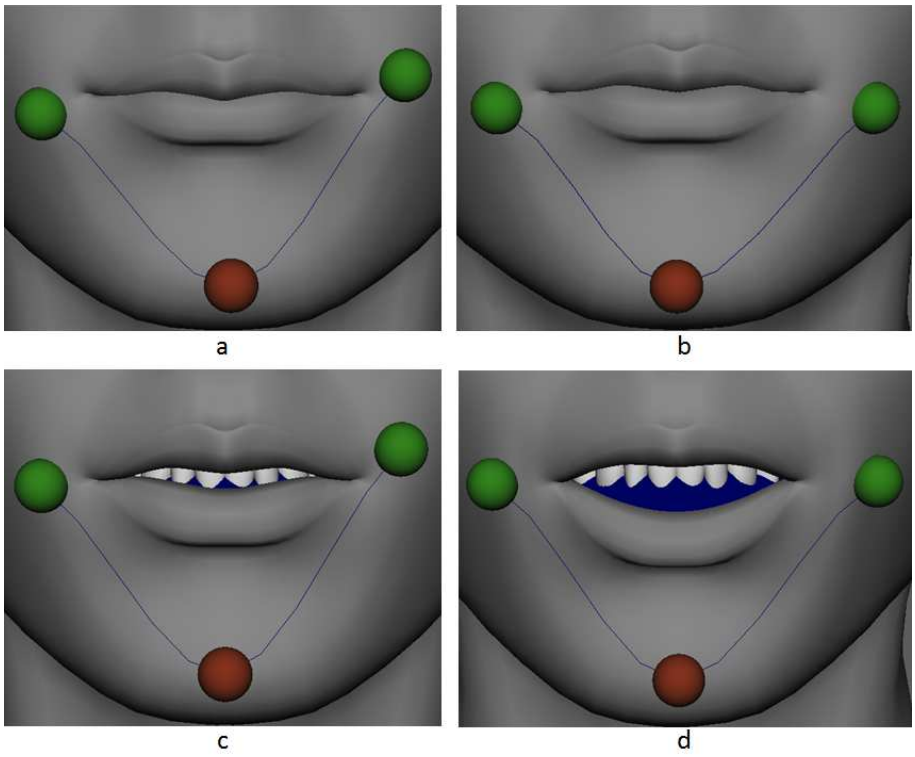

Figure 3.9: Example of mouth movements when the middle pin moves to the same location under different α values (a) $\alpha = 1.0$ (b) $\alpha = 0.1$ (c) $\alpha = 0.01$ (d) $\alpha = 0.0001$

3.3.2 Localization

In our experiments, the major facial model includes 40 blendshape targets. If the user applies the blendshape direct manipulation without our localization improvement, the results may include some peculiarities. Figure 3.10 exemplifies this problem. When the operator drags the pin in the mouth region for the first time, the system may respond with an expected pose. However, at the same time, the system automatically assigns undesired weight parameters to the beneath blendshape targets. Therefore, after the user applies another pinning operation to the eyebrow area, the model responds with

unexpected visual results. Figure 3.9 shows the manipulation area is on the right eyebrow, however, both mouth and left eyebrow deforms from the drag operation in that right eyebrow. The slider interface, thereby, clearly shows almost all weights are parametrized without any user intention.

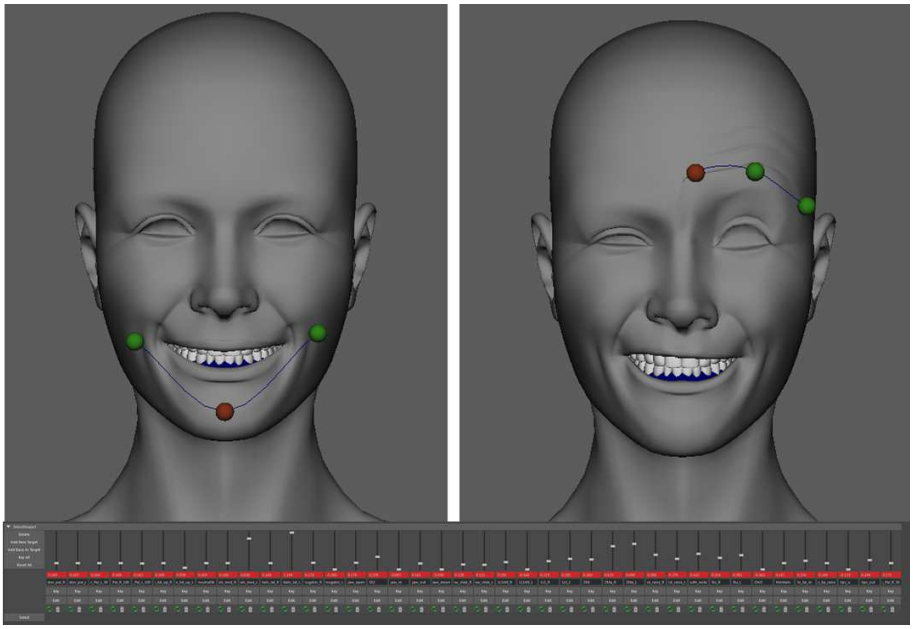

Figure 3.10: An example pose created without localization. In left image, sketching is applied to the mouth region, and a desired deformation is obtained according to the sketch alignment. However, after this sketch, almost all blendshapes are parametrized unintendently. Therefore, after sketching to the right eye-brow, the left eye, left eye-brow and mouth is deformed unexpectedly. Besides, the unintended parametrization is obvious in the slider interface. Almost all blendshapes are assigned to a weight value, which causes unintuitive and unexpected resultant pose..

In figure 3.11, we mimic the same scenario as shown in figure 3.10 to show how our localized method reacts under the same conditions. The system does not give a global deformation impact. Each pin is totally dedicated to the confined regions which enables intuitive manipulations without any undesired effects. As shown in figure 3.11, when the pin drag is applied to the right eye brow of the model, the rest of the model remains same as before. Slider interface is another proof to show the difference of our localized approach. Most of the weight values remains zero and only the corresponding weights are parametrized according to the pin drag. The images in figure 3.10 and figure 3.11 have been composed with extremity not only to demonstrate the superiority of our approach but also to analyze the resultant weight values. One significant point is to note that, equation (3.5) tends to produce negative weights or overdose weights with or without localization. Negative weights occur only in the reverse direction of the deformation and overdose weights simply mean the deformation is more than

it should be. This result mostly depends on the drag range of the manipulator. In figures 3.10 and 3.11, we set the weight range between [-10,10] only to present this effect, however most of the resultant weights stay in the range of [0,1], which the professional packages offer in their slider interface. Some of the weights, nevertheless, stay as negative and one weight stays overdose, which is shown in the comparison chart in figure 3.12.

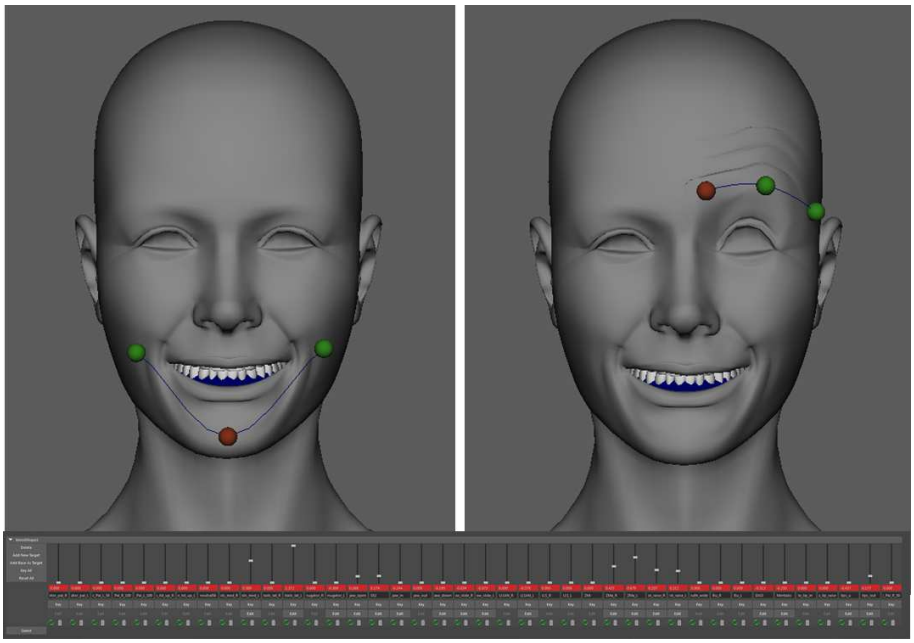

Figure 3.11: The same example pose with the previous image with localization. The same scenario is sketched as in figure 3.10. In left image, mouth area is sketched, and a desired deformation is obtained according to the sketch alignment. After the right eye-brow is sketched in the right image, the rest of the face remains the same. With our localization method, sketching only deforms the right eye-brow. Besides, it can be seen clearly in the slider interface, only the relevant blendshapes weights are parametrized. The resultant pose is more intuitive and expressive than the previous example.

Although the professional tools, such as Maya, allow to apply negative or overdose weights, the common slider interface of Maya provides the deformation space for blendshape models in the range of [0,1]. To provide a compatibility with Maya's blendshape sliders, a clipping operation is applied to the resultant weights from equation (3.5) for keeping the weights between [0,1]. The clipped weight values of the previous example are revealed in another comparison chart shown in figure 3.13. As a result, our localized technique keeps irrelevant weight values in their initial state. However the direct manipulation method without localization parametrizes these undesirable weights with random values. When the weight values given in figure 3.13 are applied to the slider interface manually, the values which are obtained from the "without localization" give hazardous and undesired results, but the values from

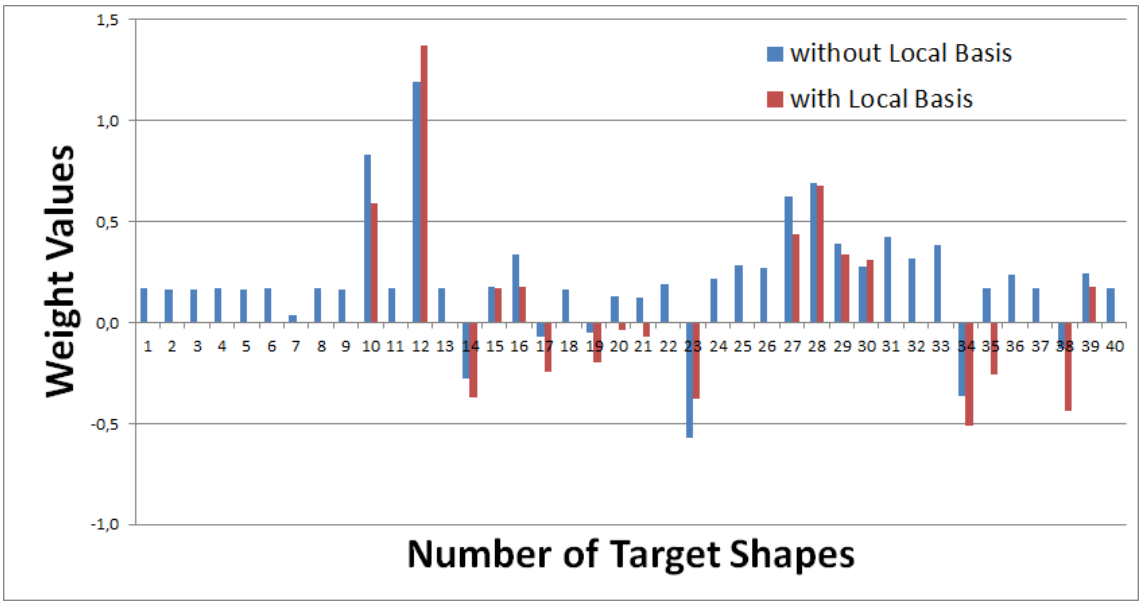

Figure 3.12: Comparison chart of the weight values for the direct manipulation with and without localization. In this chart weight range is set between [-10,10]. Direct manipulation without localization produces many non-zero weights in irrelevant blendshape targets (blue colored bars). On the contrary, our localized method just produce non-zero weights in the blendshape targets that are meant to be manipulated. In both cases, direct manipulation produces many non-negative weights, few negative weights and overdose weights.

our method give a meaningful expression to the model (see figure 3.14).

3.3.3 Manipulation with Horizontal Curves

Instead of "pin and drag" operation for the blendshape manipulation, our horizontal curve technique undertakes the manipulation in a more practical and intuitive perspective. In our interface, when the user locates the pins by sketching, they are all generated with green color. After a user selection applies to the pin for fine tuning, it immediately turns to red color. Figure 3.15 demonstrates example facial poses automatically generated by using our sketched curves. All pins remain in green color which proves that pin-and-drag operation are not applied manually. However, manual drag is inevitable to obtain the desired pose and our system allows the user to select any pin and drag it until the desired expression obtained. In addition, our system incorporates blendshape sliders if particular details are required. Figure 3.16 demonstrates the whole operation from the neutral pose to the final result on different models.

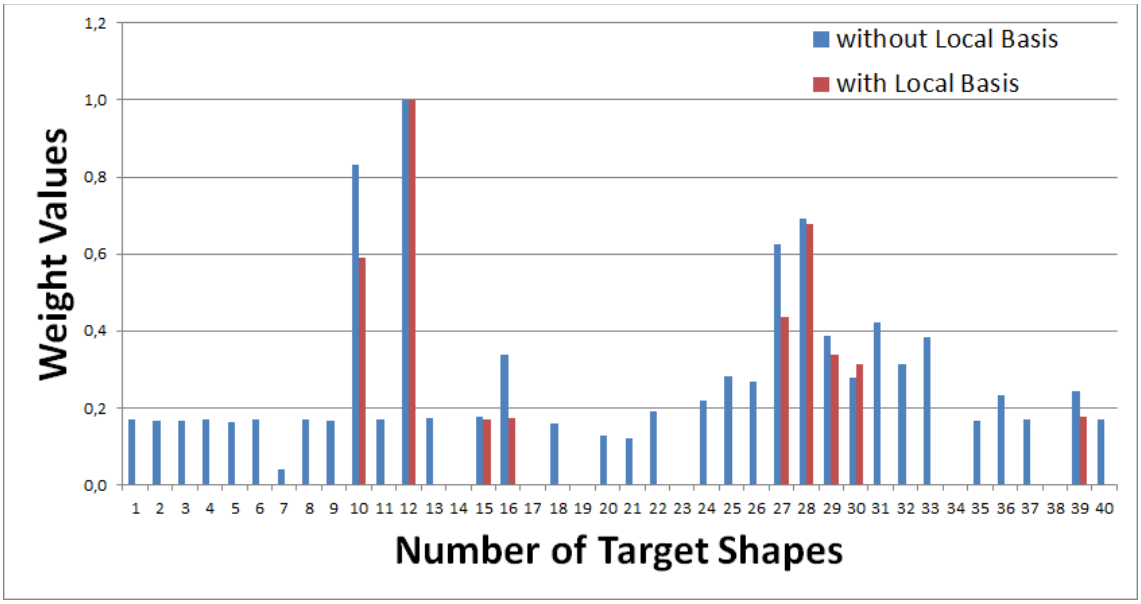

Figure 3.13: The same comparison chart of the weight values for the direct manipulation with and without localization that is illustrated in figure 3.12. In this chart weight range is set between [0,1], and the comparison is more obvious. Our localized method completely clears the irrelevant weight values that are produced by mathematical framework of direct manipulation.

3.3.4 Performance

In general, performance is a key factor in sketching systems. The user's tendency is to obtain the visual results immediately after the sketch drawing is completed. In bone-based rig models, performance is not considered as a difficulty, because the deformation space is relatively low when compared to the blendshape models. Although the main concentration of this chapter is not about speed optimization, the efficiency of our system is remarkable. Primarily the complete system has been built as a plugin for Maya, therefore our system's response time directly depends on Maya's process efficiency. Besides, mainly two factors affect the overall efficiency of our system: size of the input matrix $\bar{B}$ from equation (3.3) and the heat method for localized deformation. Size of $\bar{B}$ is directly related with the number of the target shapes of the model. Since we consider $\bar{B}$ matrix is created during the generation of each individual pin, the facial model structure is a critical element which directly affects the response time of our system. For the best performance, the model has to be cleared out from the redundant elements, which are used for creating the model during the modeling operations such as smoothing, merging, triangulation, etc. After modeling, these elements are not needed but usually retained in the model's hierarchy and seriously affect the sketching performance. Table 3.1 summarizes the differences of pin generation timings between

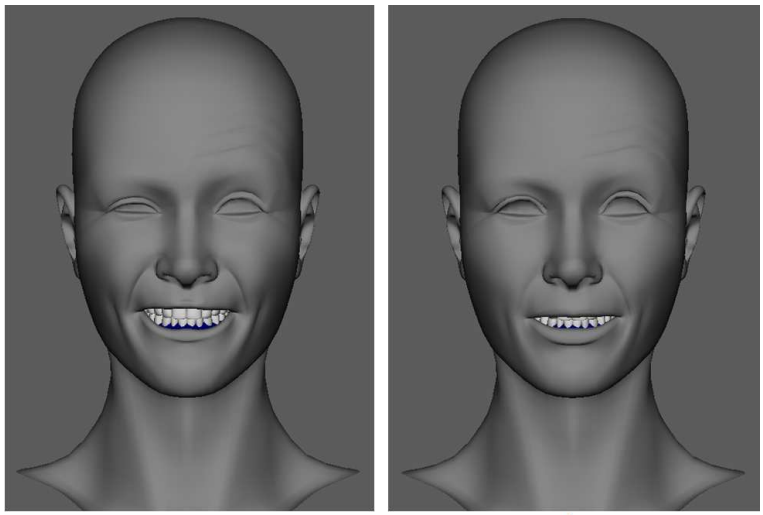

Figure 3.14: Comparison of the images with the clipped weight values. (a) shows the results which comes from "without localization", (b) shows the results which comes from "with localization"

a clean model and the same model which includes these redundant attributes. On the other hand, performance of the heat method is independent from the model structure and mostly depends on the surface components of the model (vertices, edges and faces). During the experiments, the timings for computing short range geodesic circles with heat method are steady for each individual pin generation (figure 3.4 and 3.5). For example, the heat method computes the geodesic circles around 0.02 seconds for both models: Face 1 (the model without texture) with 6720 vertices, 19984 edges and 13264 faces and Face 2 (the model with texture) with 9562 vertices, 28416 edges and 18848 faces. Finally, the timing for pose editing is around 0.01 seconds for both models. This timing is same for pose editing with either pin drag or curve entities, and it should be noted that our technique guarantees to give an interactive visual feedback by solving equation (3.5) within 0.01 seconds. The processed data, models, the method parameters and computation timings are summarized in table 3.1.

3.3.5 Animator Feedback

We have performed several informal qualitative tests with professional animators. These tests are not meant to be an exhaustive and formal evaluation, nevertheless it exposes a general information about the usability of our system. In general, the feedback has been positive. After spending some time to gain familiarity with the system, they all agreed that generating a facial pose by using our method is easier and faster than using the traditional slider interfaces. Further, one comment was

Model	$\#Vertex$	$\#Edge$	$\#Face$	$\#Targets$	α	r_{min}	r_{max}	T_{heat}	T_{pin}	T_{edit}
Face 1	6720	19984	13264	40	0.0001	0.1	0.5	0.015	0.06	0.0099
Face 2	9562	28416	18848	18	0.0001	0.1	0.5	0.02	0.015	0.0099
Face 1*	6720	19984	13264	40	0.0001	0.1	0.5	0.015	2.70	0.0099
Face 2*	9562	28416	18848	18	0.0001	0.1	0.5	0.02	1.678	0.0099

Table 3.1: Summary of the processed datasets. From left to right, columns show in order, the model used, number of vertices, number of edges, number of faces, number of blendshape targets, α value used in regularization term in equation 3.3, r_{min} and r_{max} for the range of the support region computed by the heat method, T_{heat} is average timing for heat method, T_{pin} is average timing for pin generation, T_{edit} is exact timing for pose editting. (*) means the model with "redundant attributes". All computation times are in seconds.

that using a sketch-based interface on a blendshape model allows to produce visually more plausible poses than joint-based models in [MAO+12]. Another comment was that our localization approach is intuitive and practical in terms of editing only the desired area of the face without using a confusing user-interface. Another comment stated that synchronized control of the blendshape slides has a potential to reduce the labor intensive production times. Negative comments were mostly about the detailed editing. One comment cautioned that the slider interface may still be useful in some cases to fine tune the resultant pose. A final comment emphasized that the horizontal curves are useful to rapidly create the abstract poses, but to obtain a highly precise pose requires both pin editing and slider editing together. Fortunately, our method and its interface provides the individual pin drags incorporates the underlying slider interface.

Limitations: Our method works well with blendshaped facial models however it has some limitations. At this stage the implementation of the heat method is realized only for the triangulated mesh models. The model should be constructed very accurately and carefully for obtaining the high quality visual results, because the facial poses are intrinsically limited to the span of the blendshape basis. For example, in our experiments one of the models showed a slight movement in the mouth when the edge of the eyebrow is moved (this coupling is unexpected but not necessarily unrealistic). The strokes shall not overflow to the outside of the boundaries of the model and we assume that the operator has basic artistic skills.

3.4 Summary

In this chapter, we have presented the localized blendshape direct manipulation using a sketch-based interface. Our system has primarily focused on both localized deformation and precise interconnection between the sketching and manipulators on the surface of the facial model. Our system's intuitive interface provides the user an easy and interactive approach not only to locate the pins onto the model but also to manipulate the relevant underlying blendshapes by using horizontal curves approach. When compared with other techniques, our method enables the user to create general facial expressions without using the traditional slider-based editing interface while nevertheless keeping the semantics of the blendshape model. Our localized direct manipulation method satisfies the users with its intuitive, efficient, and precise control without giving any global deformation impact to the model.

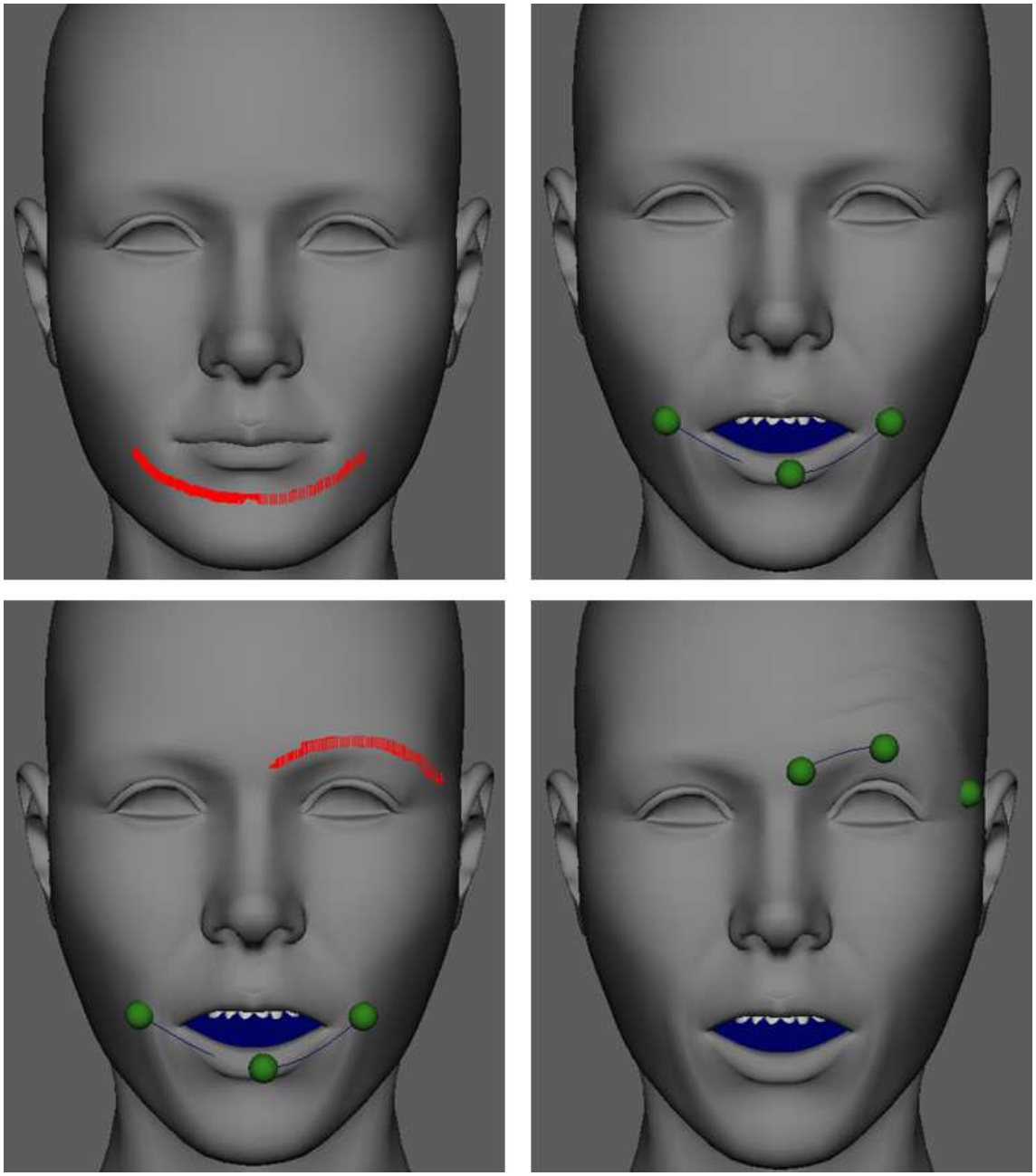

Figure 3.15: Facial pose editing on a blendshape model by using the proposed localized direct manipulation method using the sketching operations.

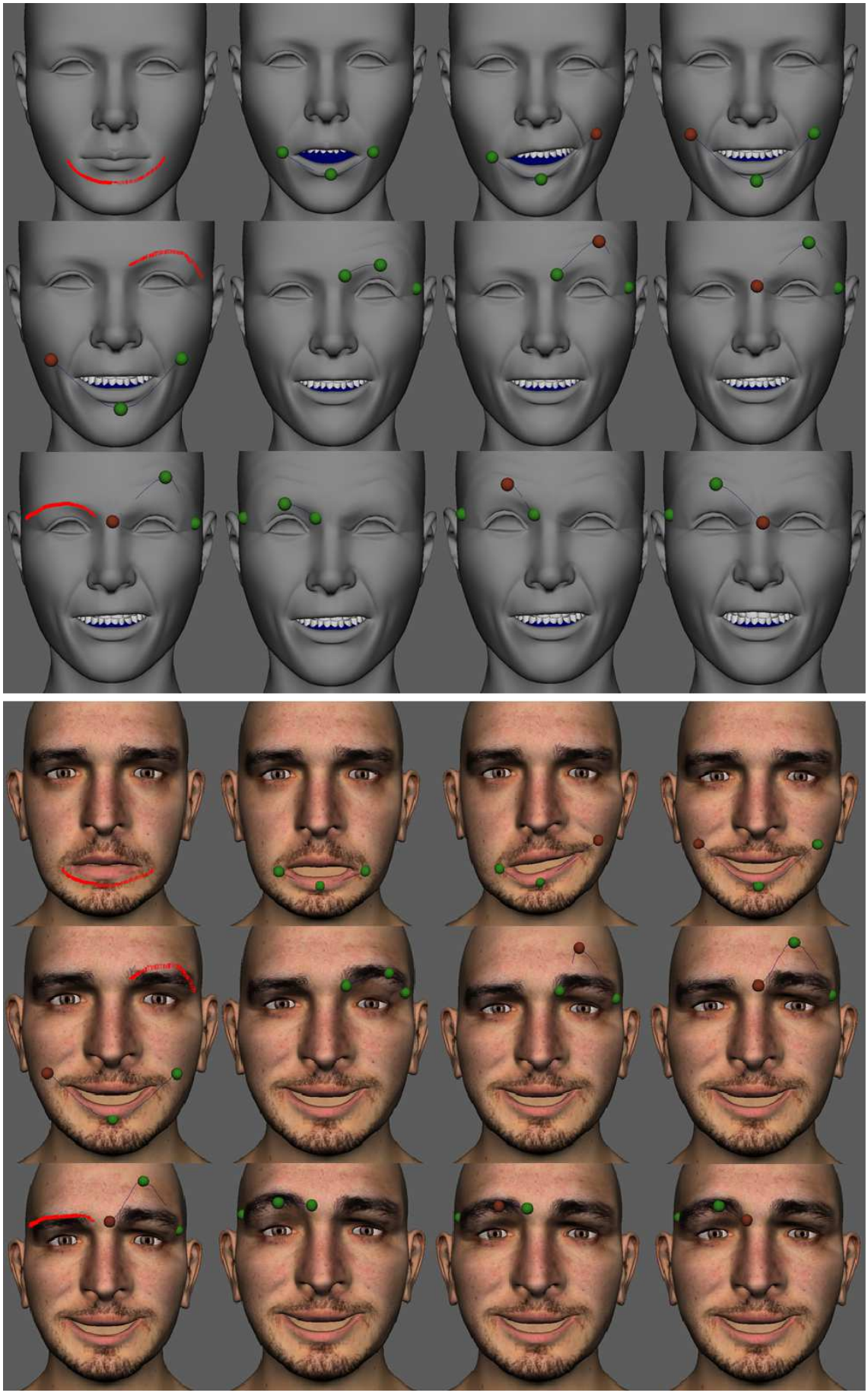

Figure 3.16: Sample sequence to create a facial pose on 2 different facial models by using our system.

Chapter 4

Transposition Based Blendshape Direct Manipulation

The mathematical frameworks of the direct manipulation techniques are mostly based on pseudo-inverses of the blendshape matrices which include all target shape's vertex positions. However, the pseudo-inverse approaches often give unexpected results during the facial pose editing process because of their unstable behavior. To this end, we propose the transposition method to enhance the direct manipulation by reducing the unexpected movements during weight editing. Our method extracts the deformation directions from the blendshape matrix and directly maps the sparse constrained point movements to the extracted directions. Our experiments show that, instead of psuedo-inverse based formulations, transposition based framework gives more smooth and reliable facial poses during the weight editing process. The proposed method improves the fidelity of the generated facial expressions by keeping the hazardous movements in a minimum level. It is robust, efficient, easy to implement and operate on any blendshape model.

4.1 Motivation

According to blendshape direct manipulation [LA10], moved points determine the resultant face as a simple linear combination of predefined target shapes,

$$m = Bw \tag{4.1}$$

where m is the vector of resulting vertex positions that are already moved, w is the corresponding weight vector and B is the blendshape matrix which includes vertex coordinates of all target shapes at its columns. The solution of equation (4.1) shows that there is an obvious pseudo-inverse relationship between the blendshape weights and manipulated points. The general pseudo-inverse approach gives the best possible approximated solution to w by minimizing the moved point to its correspondent target shapes ($\min_w \|Bw - m\|^2$). However, pseudo-inverse based mathematical frameworks are known with their unstable and unintuitive results. Altering an arbitrary side of the face model causes unexpected little changes in the remaining parts of the model. The reason for this instability is the fact that pseudo-inverse tends to assign non-zero values to its columns which cause unintended moved point projections for the irrelevant weights. To prevent the weight editing system from the unexpected movements, we offer the transposition based direct manipulation for the blendshape models. Our method is inspired from the *Jacobian Transpose* method and provides the direct projection of the moved points onto B which avoids the instability of the pseudo-inverse approach. During weight updates, keeping B^T with its original values allows us to manipulate desired part of the face without any little alterations in the remaining parts of the 3D model (see Figure 4.1).

In terms of computing the weight updates directly using the transpose of B, we analyzed the deformation (vertex displacements) directions to prove that our transposition based direct manipulation is a modified version of the classic pseudo-inverse approach. By considering each column of B is a deformation vector for the base neutral pose, we extract the deformation directions from the columns of B by using *Gram-Schmidt* process and store these columns as a new orthonormal matrix which shares the same deformation directions as B. After applying the pseudo-inverse algorithm to the new matrix, the resultant mathematical framework has appeared similar to the proposed transposition based framework with an additional step-size matrix.

In our experiments, we compare the proposed method with two different pseudo-inverse based algorithms for blendshape weight updates. The first version is the classic pseudo-

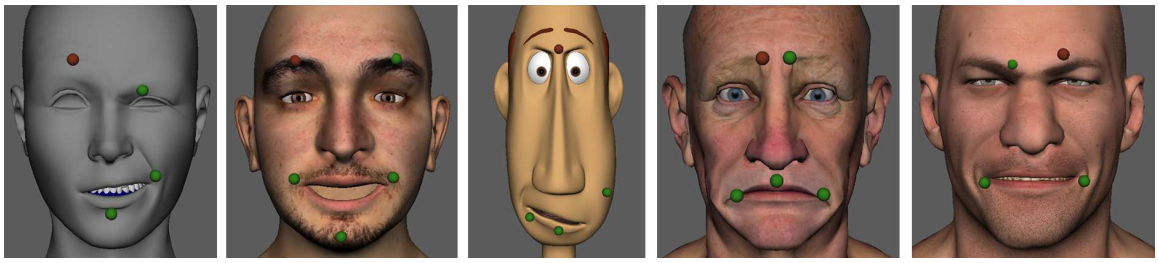

Figure 4.1: Face posing with our novel transposition based blendshape direct manipulation method. Unlike the pseudo-inverse based techniques, our method generates the facial poses without any artifact or unexpected movements. Our method provides the artistic editing with a simple, efficient and easy-to-implement mathematical framework which is inspired by *Jacobian Transposes*.

inverse method and the other one is the commonly used pseudo-inverse with *Tikhonov* regularization [LA10]. Against both methods, our method produce more intuitive and reliable posing results. Further, we discuss the advantages and disadvantages of both approaches and present a hybrid approximation by using only powerful sides of pseudo-inverse based approach and our transposition based method. Finally, the resulting approach is simple to implement, efficient (based on solving only linear systems) and can be adapted easily to the existing blendshape deformation frameworks.

4.2 Method

In this section, we first explain the algebraic perspective of blendshape models and pseudo-inverse approaches and then describe about the details of the proposed transposition based direct manipulation.

4.2.1 Blendshape Direct Manipulation and Pseudo-Inverses

According to direct manipulation scheme, the weight updates can be simply approximated as in equation 4.2:

$$w = B^{+}m \tag{4.2}$$

where B^{+} denotes for the pseudo inverse of B. B^{+} is thereby defined as $(B^{T}B)^{-1}B^{T}$. However, in some cases the inverse operation in equation 4.2 may fail because of the singularity of $B^{T}B$. In these cases, blendshape matrix may not provide a full rank.

Lewis and Anjyo [LA10] addressed this problem with the *Tikhonov* regularization term to update the weights with the most optimal fit to the point movements. Another advantage of applying regularization term is keeping the equilibrium of the system by fitting the new weight values according to the desired pose during point movements. According to [LA10], equation 4.3 shows the weight update after applying regularization to the system defined in equation 4.2.

$$w = (B^T B + \alpha I)^{-1}(B^T m + \alpha w_0) \tag{4.3}$$

where α is the regularization term which is defined by user. As mentioned in the previous chapter, α term is a small number such as 0.0001. After setting the α term to a small value, the term αw_0 in equation 4.3 does not have a noticeable effect to determine the final weights. If the regularization term is selected as zero, then equation 4.3 transforms to the classic pseudo-inverse form which is shown in equation 4.2. While dragging the constraint points on the surface of the model, equation 4.2 or 4.3 constantly updates the weight values for each coordinate unit. This approach has been widely employed for the last decade.

4.2.2 Transposition Method

Direct manipulation formulation is based on pseudo-inverse of B with or without regulation term. However, pseudo-inverse approaches have a well-known instability problem. Therefore, unexpected and unintuitive movements can be observed during the pin drag. For example, while the user drags a pin in the mouth region of the face, eye brows may move reluctantly. One view of the underlying cause of this instability is the fact that pseudo-inverses tend to apply non-zero values in its columns. Since we consider the column space of pseudo-inverse is the major deformation space of the linear system, each movement of pin, thereby, gives a global deformation impact to the facial model.

Inspired from the Jacobian transpose approach of differential inverse kinematics [LD10], we reformulate the blendshape direct manipulation in a more stable and efficient perspective. The inverse kinematics problem is explained as $\min_q \|p - f(q)|^2$ which was solved by [ZB94] by using a nonlinear programming. According to their approach, nonlinear $f()$ turns into a linear function about the current position with an update equation:

Figure 4.2: (a) Neutral pose of the model with the located pin. (b) Pin moved down and right by using our "Transposition Method". (c) Pin moved to the same position as (b) by using the "Pseudo-Inverse Approach".

$$\dot{p} = J\dot{q} \tag{4.4}$$

where $\dot{p}$ is the vector from the point on the inverse kinematic handle to the target point, J is the Jacobian of $f()$. Equation 4.4 has to be solved for the parameter update $\dot{q}$ with the most optimal approximation. The resultant system is nothing but $\dot{q} = J^{+}\dot{p}$. The problem in equations 4.1 and 4.4 nicely fits to each other. However, Welman [Wel93] offers the *Jacobian Transpose* algorithm for differential inverse kinematics to avoid the unstable behavior of pseudo-inverse approach. According to this algorithm, the parameter update can be defined $\dot{q} = J^{T}\dot{p}$ instead of $\dot{q} = J^{+}\dot{p}$. In our transposition method, we adapt *Jacobian Transpose* algorithm to the blendshape direct manipulation in an elegant way to reduce the global deformation impact to a minimal level. Therefore, the weight updates are defined as:

$$w = B^{T}m \tag{4.5}$$

The transition from pseudo-inverse based weight updates to our transposition based method is given in Appendix C. From an intuitive perspective, equation 4.5 provides a direct projection of the point movements onto the B. Therefore, point displacements directly determine the exact deformation with the correcting rows of B^{T}. This type of approach prevents the unexpected and unintuitive deformation effects on the facial model. We accomplish this by appending the constrained points to m and corresponding rows to B. Thereby, each pinned points adds three columns to B^{T} and rows to m.

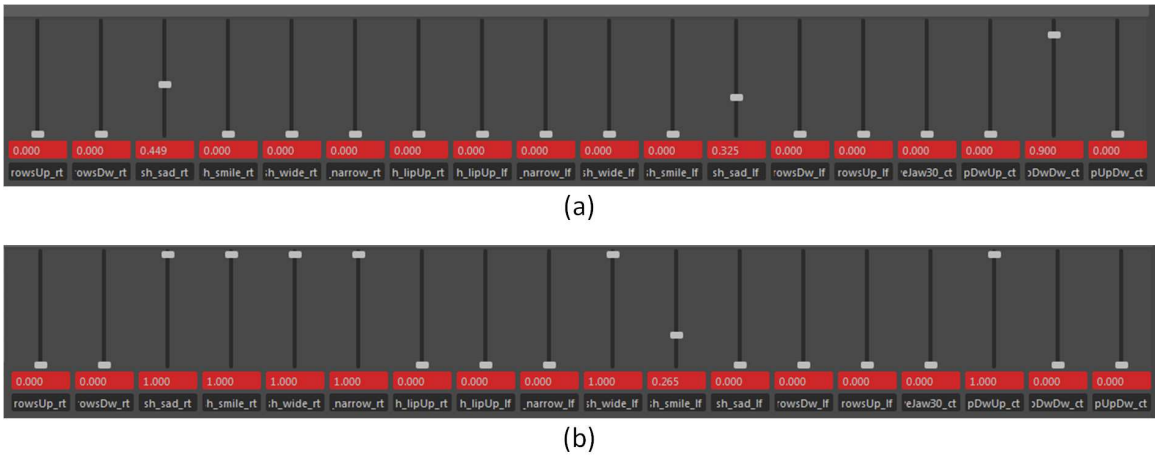

Figure 4.3: The screen-shots of the slider interface according to the pin movements that is shown in figure 4.2. (a) is for our "Transposition Method". (b) is for the "Pseudo-Inverse Approach".

For example, while "n" arbitrary number of pins are located on the 3D facial model with "k" number of blendshape targets, the system described in equation 4.5 can be represented in the matrix form in equation 4.6:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_k \end{pmatrix} = \begin{pmatrix} x & y & z & \cdots & z \\ x & y & z & \cdots & z \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{b_1^T} & \mathbf{b_2^T} & \mathbf{b_3^T} & \cdots & \mathbf{b_{3n}^T} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} m_{1x} \\ m_{1y} \\ m_{1z} \\ m_{2x} \\ m_{2y} \\ m_{2z} \\ \vdots \\ m_{3nz} \end{pmatrix} \tag{4.6}$$

It should be noted that in equation 4.6, B^T is $k \times 3n$ matrix consists of the coordinates of blendshape targets over the selected vertex and w vector which consists of all weight values that form the resultant face. It is clear that $w_i = b_i \cdot m$ from equation 4.6. Therefore, this proves each pin displacement is only projected on its corresponding blendshape targets and this behavior keeps the system stable, intuitive and prevents the unexpected movements.

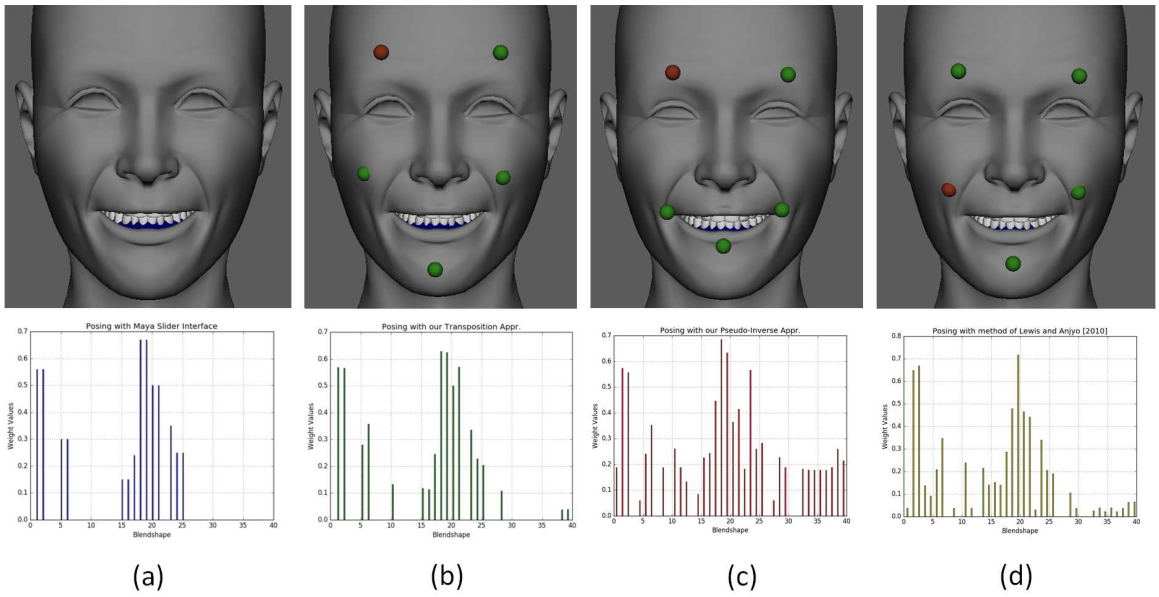

Figure 4.4: Blendshape direct manipulation comparison example on a model with 40 blendshape targets. (a) is the target pose which was created in maya slider interface. An experienced user generated the most closer pose to (a) by using our transposition method (b), classic pseudo-inverse approach (c) and the method of Lewis and Anjyo [LA10]. Below, the weight values are reported for each above corresponded pose. Our transposition based approach clearly gives the most close approximation to the target pose.

4.2.3 Pseudo-Inverse with Transposition Method

The pseudo-inverse and blendshape transpose approaches have their own complementary advantages and disadvantages. Therefore, it would be desirable to combine them and obtain a smooth and promising hybrid approach. The instability problem of pseudo-inverse approaches are already mentioned in the previous section. However, this negative effect can be used as an advantage during the pin movements. Because in some particular cases, the movements of shapes naturally requires the movements of other shapes. For example, the specific movement of inner eyebrow for the "anger expression" can only be possible with the slight movement of the mouth. During pose editing, this type of coupling may seem unexpected but not necessarily unrealistic. Besides, the gross pin movements may overshoot the pose with the transposition method. By taking these possibilities into account, we define a simple parametric representation for the weight updates that is shown in equation 4.7:

$$w = (\gamma B^{+} + (1 - \gamma)B^{T})m \tag{4.7}$$

where γ is an arbitrary scalar which keeps the balance of the system. The value of γ would be varies between [0,1] where error stay in a minimal state. Unfortunately, in this context there is no exact definition of error, since pseudo-inverse part of equation 4.7 already produces a slight error. The error, nevertheless, can be considered as the unexpected movements of the face that are beyond the tolerable limit. For the most optimal fit, we suggest to keep the value of γ as a small number such as 0.1. The further details of equation 4.7 and the γ value will be discussed in the next section.

4.3 Validation and Experimental Results

Our method and its interface have been implemented as a plugin for Maya 2016 using Python. Numpy package has been used for matrix calculations. We demonstrate the comparison of our transposition method with the other existing techniques on the various blendshape facial models to validate the stability of our method. All direct manipulation techniques including our transposition method respond in real time and rapidly produce the interactive visual feedbacks to the user. Our main motivation has been keeping the user away from the technical aspects of the method during the implementation. For that reason except the mouse, the interface does not include any control components. All the tests and validations were performed on 4-core Intel Core i7-2600 3.4 GHz machine with 8 GB of RAM and an nVidia GTX 570 GPU.

Comparison to pseudo-inverse based approaches

Despite the instability problem of pseudo-inverse based approaches, over years they have dominated the research on direct manipulation. However, our experiments show that our transposition method deforms the desired area of the face with a more smooth blending. Besides, the unexpected movements are almost completely eliminated. The figure 4.2 visually compares our method with the pseudo-inverse technique under the single pin editing scenario. The model consists of 18 blendshape targets and the same pin movements produce thoroughly different facial pose. The slider interface of figure 4.2 is shown in figure 4.3. The pseudo-inverse approach clearly updates the least corresponding weights more than it should be, and the resultant pose stays far away from the expected pose. On the other hand, our transposition method directly maps the displacement of the pin movement onto the corresponding blendshape targets. The resultant pose is in the direction of the expectations. Besides, it should be noted

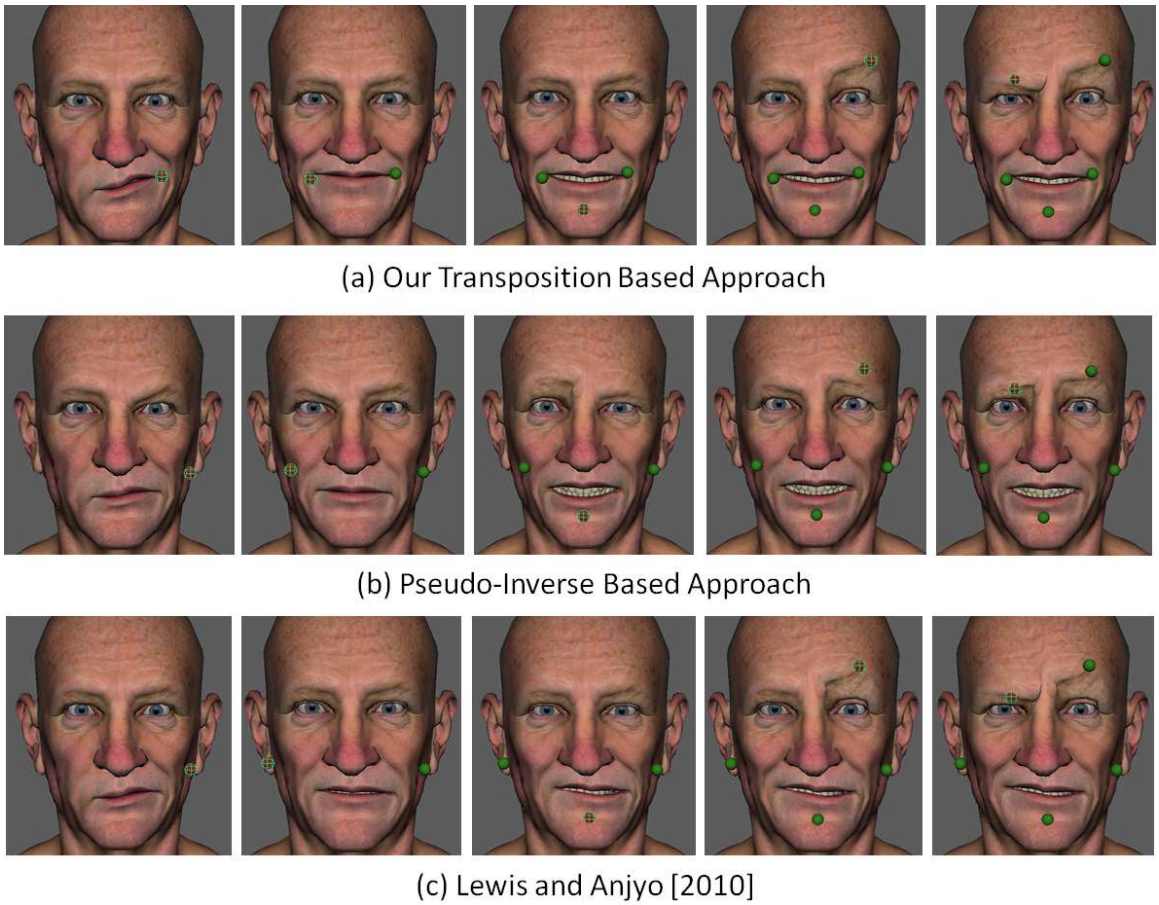

Figure 4.5: Comparison of our proposed method with other pseudo-inverse based techniques on a production quality blendshape model with more than 100 target shapes. Our transposition based method (a) generates visually plausible and desired results with same pin drag direction. The other methods (b and c) produce unintuitive results with many unexpected global deformation.

that both approaches operate in the deformation space defined by the predefined blendshape targets. Therefore, it is not possible to express poses beyond the limits of what blendshapes allow to the user.

Further, we created an experimental pose (figure 4.4a) for an experienced user and asked briefly produce the same pose using our transposition based method, pseudo-inverse based approach and method of Lewis and Anjoy [LA10]. The test model consists of 40 blendshape targets and we measured each target shape value as shown in figure 4.4 below. The most correct pose was intuitively obtained by our transposition method. However, all 3 approaches produce errors. These errors are reported using the root mean square deviation per weight:

Model	$\#Vertex$	$\#Edge$	$\#Face$	$\#Targets$	T_{pin}	T_{edit}
Face 1	6720	19984	13264	40	0.15	0.0099
Face 2	9562	28416	18848	18	0.03	0.0099
Face 3	11926	35576	23648	36	0.13	0.0099
Face 4	5692	11282	5584	127	1.75	0.015
Face 5	5792	11625	5834	136	1.46	0.019

Table 4.1: Summary of the processed datasets. The face model order follows the order in figure 4.1 (from left to right). For example, Face 1 is the left most model, Face 3 is the cartoon model in the middle, Face 5 is the right most model. In the table from left to right, columns show in order, the model used, number of vertices, number of edges, number of faces, number of blendshape targets, T_{pin} is average timing for pin generation, T_{edit} is exact timing for pose editing. All computation times are in seconds.

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |w_i - \bar{w}_i|^2} \tag{4.8}$$

where N is the number of target shapes, w_i is the weight value of the i^{th} target shape, and $\bar{w}_i$ is the corresponding weight value of the i^{th} target shape of the experimental pose (figure 4.4a). The error values for each target shape is compared in figure 4.6. According to the error analysis, the pseudo-inverse approach and the method of Lewis and Anjyo [LA10] produce several irrelevant weight values during the pose generation. Besides, both approaches generate error values on the relevant weights which causes unintuitive end results. On the other hand, our transposition based method produce very few error values which mostly occur on the relevant weight values. This difference is due to precise hand gestures and almost not observable on the model. Figure 4.6 demonstrates these error differences in scatter and line graph styles.

We compare all three approaches for posing with a production quality model (number of blendshapes > 100) in figure 4.5 which is a more realistic scenario for the artists to create facial expressions during the animation process. We observe that method of Lewis and Anjyo [LA10] (figure 4.5c) produces visually more plausible poses that the classic pseudo-inverse approach (figure 4.5b), nevertheless they both produce unexpected movements on the other parts of the desired parts of the face. Our transposition based method (figure 4.5a) provides almost fully local deformation which allows the artists to edit only the desired manipulation area without any visible global deformation impact. It should be noted that in figure 4.5, all columns are aimed to produce the same pose with the same number of pins. Besides, figure 4.8 illustrates the application of our transposition based method on different facial models. Table

1 summarizes the processed data, model details and computation timings for the proposed approach.

Parametrization of the Hybrid Approach

Assigning a parameter to γ value in equation 4.7 is the most straight forward approach. It would be most likely to choose a small heuristic value (such as 0.1) and allow the transposition method to dominate the equation 4.7. Instead, we apply the *singular value decomposition* to B^T where $B^T = VSU^T$ and choose the smallest singular value σ_1. The singular values smoothly approaches to zero when the number of pins on the model increase. On the other hand, since $B^+ = VS^+U^T$, the singular values of pseudo-inverses grow toward infinity.

As the user applies few number of pins to the model (such as 1 or 2), the pseudo-inverse side of the equation 4.7 will be intrinsically effective. Besides, by intuition few number pins denotes that the user does not fully constrain the model and execute only a simple weight update. Another possibility can be an unstable pose needed for a particular case. Nevertheless, the general practice is based on applying the high number of pins to the model (such as 5 to 15) and controlling all of them without any artifact. In these cases, our transposition method dominates the equation 4.7 and provides smooth results. Figure 4.7 demonstrates an example case to the usage of equation 4.7. Some poses are necessary for irregular expressions and possible to create with 1 or 2 pins similar to first and second column of figure 4.7. In these cases, pseudo-inverse approach can be considered to generate the desired irregular pose. However, when the number of pins increase on the model, transposition based method dominates the equation 4.7 (last column of figure 4.7).

4.4 Summary

In this chapter, we present a practical transposition based blendshape direct manipulation method for 3D facial models. Our method has focused on the precise projection of the pin displacements onto the corresponding deformation vectors. Unlike the previous pseudo-inverse based methods, our method allows direct control of the deformation space by purely sticking its dimensions. This behavior reduces the unexpected movements during the artistic editing and almost completely reduces the deformation influence area to the local geometry where the pins are located. Besides, we offer

a hybrid approach which incorporates our proposed method with the pseudo-inverse solution. In the results, we demonstrate how our method provides intuitive and artistic edits on blendshape models almost without any artifacts. Our method is original, simple and general that can be easily adapted to the existing direct manipulation frameworks.

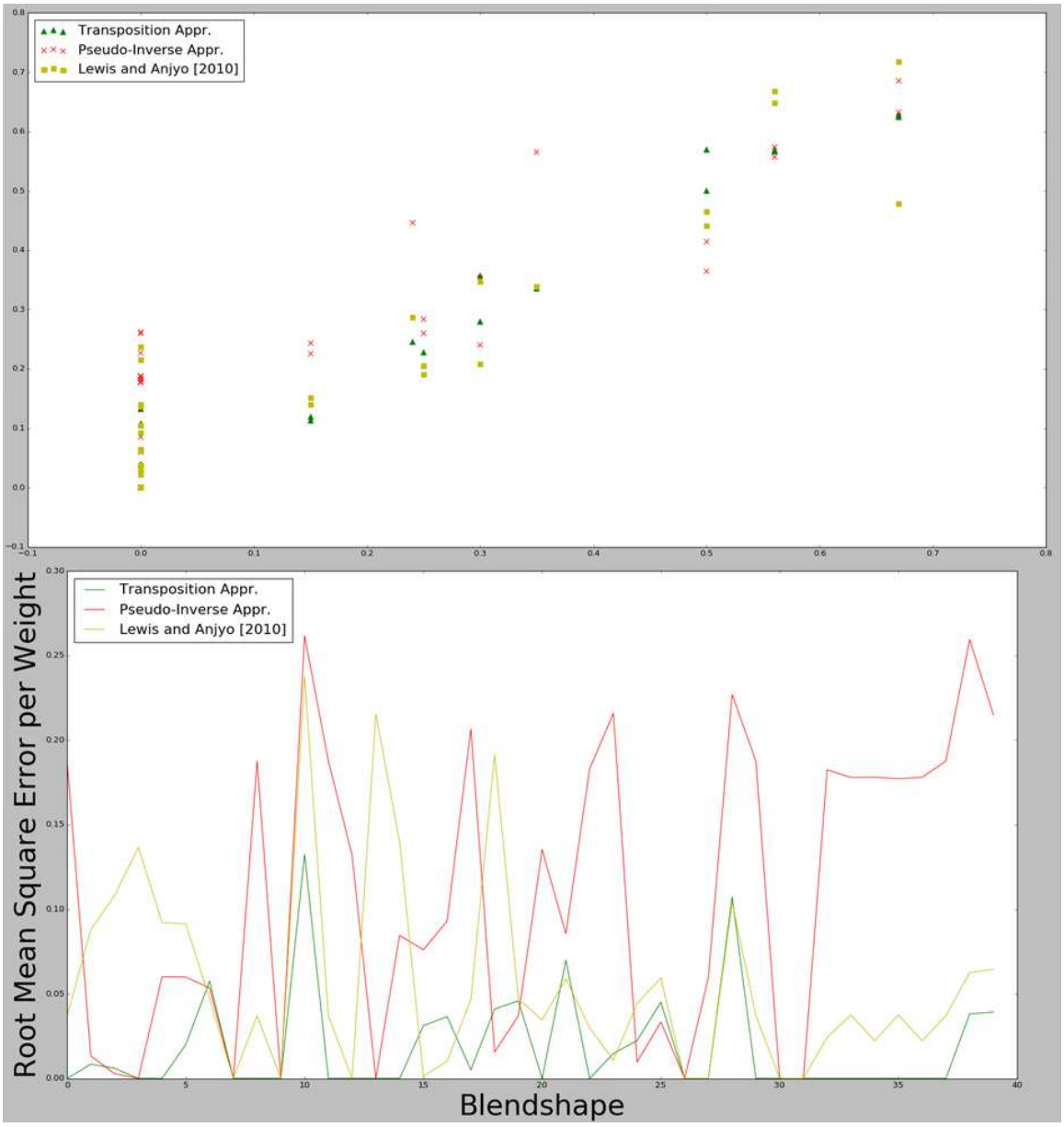

Figure 4.6: The error analysis of the weight deviation from figure 4.4. The error values are calculated by using equation 4.8. The scatter plot of error values are presented in the top graph and the comparison of error values are shown in the bottom graph. Our transposition method produces the minimum error during the pose generation.

Figure 4.7: The example illustration of our hybrid approach from equation 4.7. The parametrization is calculated according to singular values of SVD. When the user locates few number of pins on the model, SVD produce a high value for the *gamma* parameter and pseudo-inverse side of equation 4.7 dominates. On the other hand, with high number of pins on the model, SVD produces a very small number and transposition part of equation 4.7 dominates. In some rare cases, our hybrid can be useful.

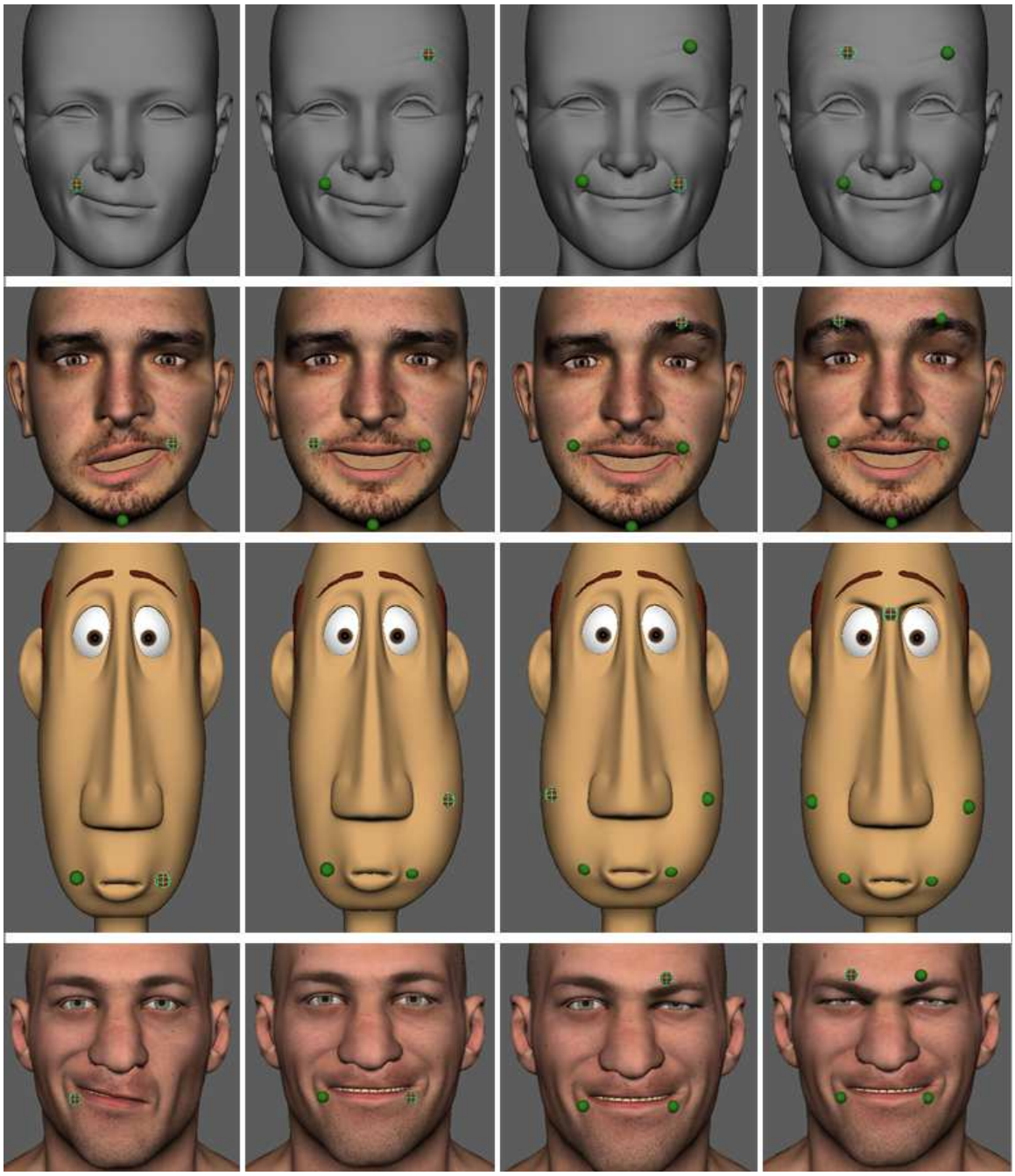

Figure 4.8: Sequential posing using our transposition based blendshape direct manipulation method. Our method can be applicable to all blendshape models without any problem and it took less than 3 minutes to create the final expression from the neutral pose. The details of the models are reported in table 1.

Chapter 5

Localized Verlet Integration Framework For Facial Models

Traditional Verlet integration frameworks have been successful with their robustness and efficiency to simulate deformable bodies ranging from simple cloth to geometrically complex solids. However, the existing frameworks deform the models as a whole. We present a Verlet integration framework, which provides local surface deformation on the selected area of the mesh without giving any global deformation impact to the whole model. The framework is specifically designed for the facial surfaces of the cartoon characters in 3D computer animation. Our framework allows an interactive selection of the deformation influence area over the model, which gives a notable benefit in terms of efficiency for the position manipulation during the simulation process. To this end, we take advantage of a geodesic distance computation technique based on heat kernel for the determination of the selected surface on the model. In addition, the framework exploits the projected position constraints for stretching, shearing and bending to handle the environmental interactions such as collision. The proposed framework is robust and easy to implement since it is based on highly accurate geodesic distance computation and solving the projected geometric constraints. We demonstrate the benefits of our framework with the results obtained from various facial models to present its potential in terms of practicability and effectiveness.

5.1 Motivation

Creating appealing surface deformation of rigid and soft bodies has been always a challenging problem for the area of computer graphics. The major reason is the increase of the expectations for more realistic effects. In parallel to that theoretical approaches to create these effects have become complicated over many years for gaining the life-like components. Although there exists sophisticated theories, which define physical laws for animation, the tendency of seeking the efficiency, easy implementation and robustness is always the main goal during research. Notable examples are Position-Based Dynamics (PBD) [MHHR07] and Nucleus [Sta09], which define objects as a particle system and particles in the system are related with each other by geometrical constraints. During the simulations, each constraint is solved iteratively in a non-linear Gauss-Seidel form.

The proposed framework in this chapter follows the similar approach as PBD. However, our framework is focused on the facial applications so we use the Verlet integration scheme from [Jak01]. The velocities are stored in an implicit fashion instead of explicit and our main focus is more centered on the direct position manipulation.

PBD approaches are quite common for cloth, deformable solids and rigid body simulations. In these particular cases, the dynamic solvers give a global deformation to the object, because the forces or environmental impacts (such as collision) naturally influence the object as a whole. The most important feature of our framework is to create a local deformation area on the model according to the needs instead of running the simulation over the whole model (or object). For that reason, we pre-compute the geodesic distance from a selected center (vertex) to a desired range by using the heat method [CWW13] and set-up a deformable soft area within the defined boundaries. After this operation, the specified deformable area has the geometric constraint properties from PBD and outside of the boundaries the model conserves its rigid behavior (see figure 5.1). This feature gives a significant benefit in terms of flexibility and control for deforming a desired region of a complex model such as face with a minor computational cost.

Our solution used in this framework provides an expeditious approach for facial surface deformation, which combines the projected geometric constraints from PBD and the heat method for the geodesic distance computation. In general, the physically-based facial rigging for deformation is a complicated and highly demanding process, which is not the focus of this chapter. We address the previously mentioned drawback of global

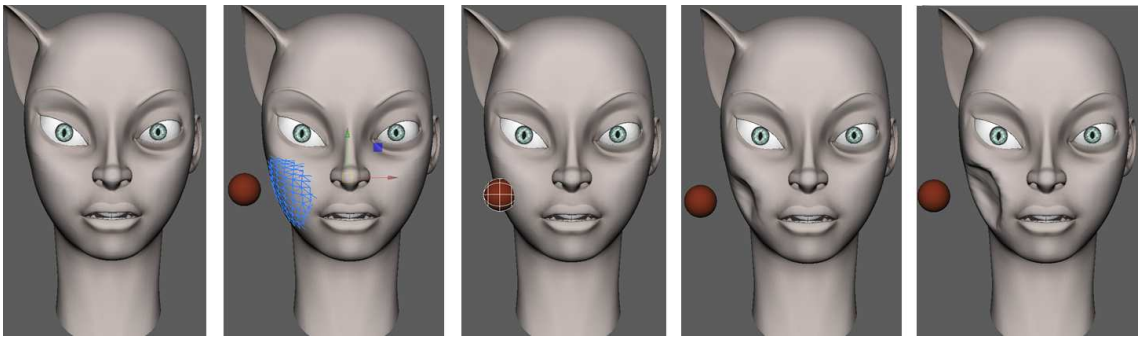

Figure 5.1: Demonstration of a simple sequence of the proposed framework on a humanoid character. The model includes 6798 vertices. After the collision detection only the assigned area from the heat method (blue painted) is deformed and the rest of the model remains same.

deformation of the complex objects by allowing the determination of the deformable surface arbitrarily on the target model. Similar to the traditional PBD systems, our framework is controllable and retains the simplicity in terms of implementation.

5.2 Position Verlet Integration

Verlet integration scheme is a refined numerical integration method for defining the equations of motion. Unlike the Euler's method [NMK+06], the main idea of this integration model is to exploit the central difference approximation to the second order differential equation shown as in equation (5.1):

$$\frac{\Delta^2 q_n}{\Delta t^2} = \frac{\frac{q_{n+1}-q_n}{\Delta t} - \frac{q_n - q_{n-1}}{\Delta t}}{\Delta t} \tag{5.1}$$

where Δt is the discrete time step, q_n is the position vector at t_n. Equation (5.1) is also the definition of acceleration ($a_n = \frac{\Delta^2 q_n}{\Delta t^2}$). According to Störmer-Verlet representation, the position updates are expressed as in equation (5.2):

$$q_{n+1} = 2q_n - q_{n-1} + a_n \Delta t^2 \tag{5.2}$$

In equation (5.2), right-hand side can be expressed as $q_n + (q_n - q_{n-1}) + a_n \Delta t^2$ which shows for the previous positions, velocities are stored implicitly [Jak01]. We additionally introduce a simple damping parametrization to equation (5.2) to ensure our framework converge to a stable and plausible state. The modified version of Verlet integration in our framework can be written as in equation (5.3):

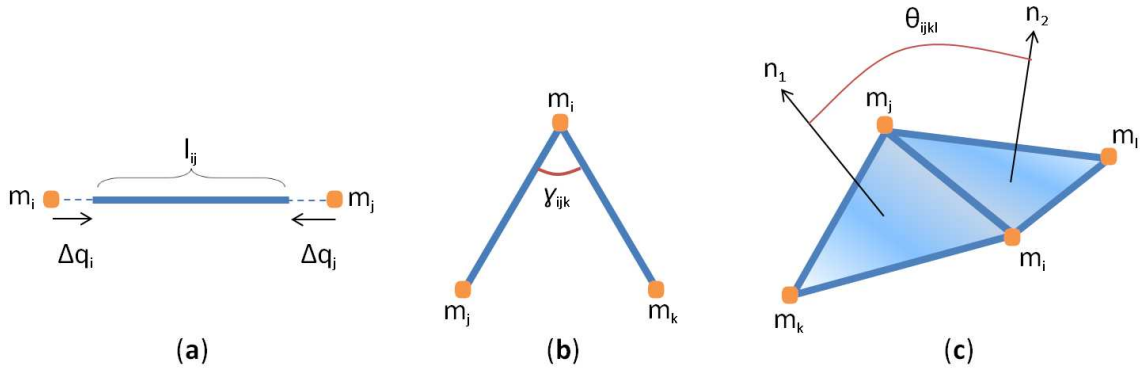

Figure 5.2: Illustrations of the constraint projections: (a) represents the stretching constraint defined in equation (5.8), (b) represents the shearing constraint defined in equation (5.9), (c) represents the bending constraint defined in equation (5.10)

$$q_{n+1} = q_n + (q_n - q_{n-1})(1 - k_{damp}) + a_n \Delta t^2 \tag{5.3}$$

where k_{damp} is a damping parameter $k_{damp} \in [0, 1]$. This slight modification to the original Verlet integration shown in equation (5.2) is a necessary step, which adds an overall aesthetics to our framework. Although damping parameter causes our framework an insignificant payoff, it is important to note that to handle more stability, damping is a required factor for the smooth and pleasing responses in our framework especially during the collision response stage.

5.3 Description of Deformation Constraints

In this section, we would like to explain the system of interconnected components of the triangulated mesh (input model), which acts like particles in position-based dynamics approach [MHHR07]. Accepting the input model as a particle system can be considered as a set points, which are connected with each other by links with an arbitrary order. This approach facilitates to operate on each component (such as vertices, edges, faces, etc.) of the input model in the proposed framework.

According to the basics of the position-based dynamics [MHHR07], [BMOT13] and [BMM15], after the predicted positions of the particles are computed in the time integration scheme, these positions are altered by a technique to match up to a group of constraints C_i. The system of constraints is iterated by a Gauss-Siedel form, which updates the positions one after another until all constraints are satisfied. During these

updates, conservation of linear and angular momentum is a significant matter, which is provided implicitly. The displacements of the particles are calculated with the solution of the following equation (5.4):

$$C(q + \Delta q) = 0 \tag{5.4}$$

where q is the concatenation of vectors, which includes the particle positions $q = [q_1, q_2, ..., q_n]^T$ and Δq is the correction displacements. Solving the constraint function, a 1^{st} order Taylor-expansion is employed to approximate, which is shown in equation (5.5):

$$C(q + \Delta q) \approx C(q) + \nabla_q C(q) \cdot \Delta q = 0 \tag{5.5}$$

Equation (5.5) is considered as an undetermined equation. Thus, the directions of the Δq is limited to the directions of the $\nabla_q C(q)$ to conserve both momentums, so with the consideration of the masses (m) of the particles, Δq becomes as in equation (5.6):

$$\Delta q_i = w_i \lambda \nabla_{q_i} C(q) \tag{5.6}$$

where $w_i = \frac{1}{m_i}$ and λ stands for the Lagrange multiplier, which obtained by substituting equation (5.6) in equation (5.5):

$$\lambda = -\frac{C(q)}{\Sigma_j w_j |\nabla_{q_j} C(q)|^2} \tag{5.7}$$

With the equations (5.6) and (5.7), the positions of the particles (or points) are updated after each constraint is computed. In our framework, 3 position constraints are chosen by following [Sta09] : stretching (C_1), shearing (C_2) and bending (C_3)(see figure 5.2). The mathematical expressions of these constraints in the same order listed as follows:

$$C_1(q_{i,j}) = |q_j - q_i| - l_{ij} \tag{5.8}$$

$$C_2(q_{i,j,k}) = cos^{-1}(M_{ij} \cdot M_{ik}) - \gamma_{ijk} \tag{5.9}$$

$$C_3(q_{i,j,k,l}) = cos^{-1}(N_{ijk} \cdot N_{ijl}) - \theta_{ijkl} \tag{5.10}$$

where l_{ij} is the rest length of each edge between the points, γ_{ijk} is the rest angle between edges and θ_{ijkl} is the rest angle between each face primitives. Besides, M and N values are computed as follows:

$$\begin{aligned} M_{ij} &= \frac{q_j - q_i}{|q_j - q_i|} \\ M_{ik} &= \frac{q_k - q_i}{|q_k - q_i|} \end{aligned} \tag{5.11}$$

$$\begin{aligned} N_{ijk} &= \frac{(q_j - q_i) \times (q_k - q_i)}{|(q_j - q_i) \times (q_k - q_i)|} \\ N_{ijl} &= \frac{(q_j - q_i) \times (q_l - q_i)}{|(q_j - q_i) \times (q_l - q_i)|} \end{aligned} \tag{5.12}$$

The types of constraints may be increased depending on the requirements. In appendix A, we derived all the constraint from equations (5.8), (5.9) and (5.10) for the gradients, Lagrange multiplier (λ) and the position corrections.

We would like to briefly mention about another constraint, which is the easy collision response realization in the proposed framework. Collision handling is a key factor for the demonstration purposes of the position constraint behaviors (see figure 5.3). We have used the advantage of generic collision handling principles in [MHHR07] and [Jak01] because of their stable comportment and smooth adaptation in our framework. Collision constraint is an inequality constraint [Sta09], [MHHR07], which makes it different from the aforementioned constraints. This unilateral behavior can be expressed as $C(q) \geq 0$. In our particular proposed case, we restricted the collision object as a simple sphere, which can be considered as a small ball. According to [MHHR07], the constraint function for static collision is defined as in equation (5.13):

$$C(q) = (q - s_{col}) \cdot n_{col} \tag{5.13}$$

where s_{col} is the surface penetration (or entry) point and n_{col} is the surface normal. Immediately after the intersection phase, the positions and velocities are updated along the direction of contact point normal at each time frame accordingly. Another

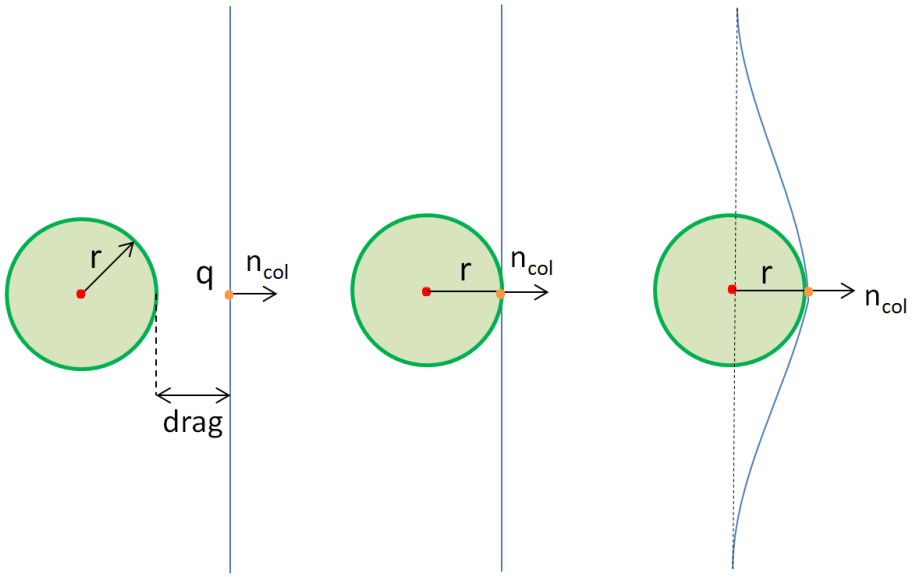

Figure 5.3: Illustration of the collision constraint: Left: Before the contact, the straight line represents mesh surface with contact point. Middle: The moment of the contact, after dragging the collision object to the surface, . Right: After the contact, the response of the surface with the continuous time.

interesting point is, the appropriate velocity updates are computed automatically during Verlet integration iterations. In our framework, collision constraint is also computed within the deformation loop of other mentioned position constraints unlike [MHHR07].

5.4 Geodesic Distance Computation

Before applying the operations mentioned in sections 4.2 and 4.3, our framework allows the selection of the desired area of mesh, which is influenced from the deformation. This gives a significant advantage in terms of locality, scalability and control (see figure 5.4). Because the direct position manipulation aspect of PBD approach occurs only inside the boundaries of the selected zone instead of the whole object. In this framework, we have chosen to compute the geodesic distances with the *"Heat Method"*, which was presented in [CWW13]. The heat method is an elegant approach to compute the geodesic distances, which is based on Varadhan's formula from 1967:

$$d(x,y)^2 = \lim_{t\to 0}[-4t \log k_{t,x}(y)] \tag{5.14}$$

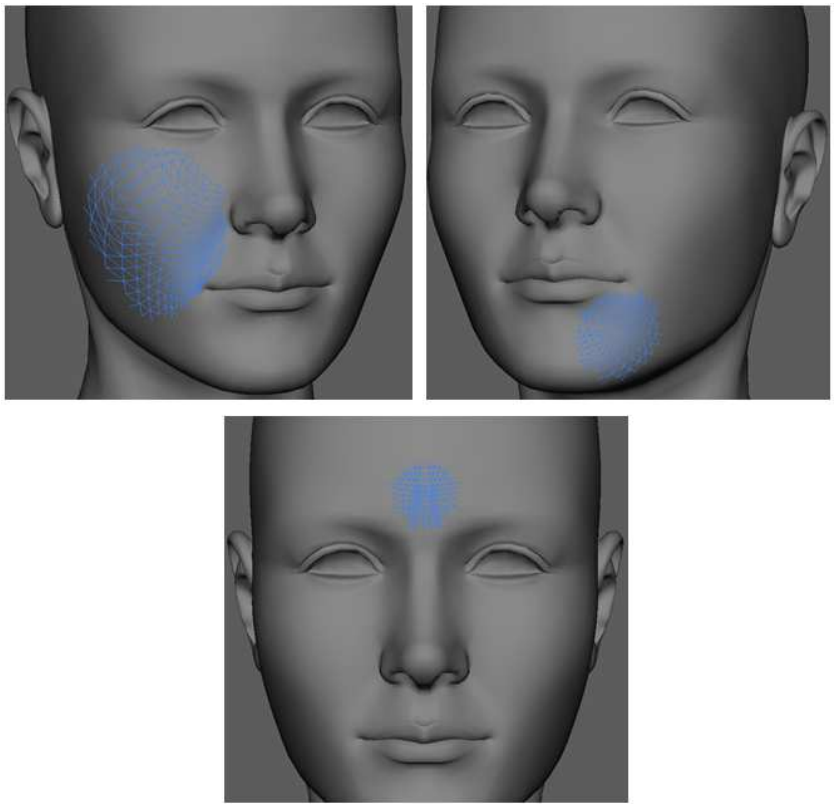

Figure 5.4: The heat method allows us to select any part of the mesh to apply the simulation. It can be scaled easily according to the desired selection. Upper Left: An arbitrary vertex in the cheek part selected with larger boundary. Upper Right: Same selection applied in the jaw with a slight narrow boundary. Down: The area selected with more narrow boundary in the forehead.

where $k_{t,x}(y)$ is the heat kernel. This formula basically explains the geodesic distance (d(x,y)) can be recovered by the heat transfer from point x to point y in a short time. The basic algorithm to the solution of finding the geodesic distances by using equation (5.14) is summarized as follows: at the beginning user-specified actions can be defined as giving the input as a triangulated facial mesh (M_f) to the framework and selecting any arbitrary vertex on the mesh ($v \in M_f$). After the vertex selection, heat method integrates the heat-flow $\dot{u} = \Delta u$ for a fixed time. After that, it evaluates the normalized gradient field $X = -\frac{\nabla u}{|\nabla u|}$. At last, it solves the Poisson equation to find the geodesic distance field (d) in equation (5.14) $\Delta d = \nabla \cdot X$. Appendix B summarizes the derivations, which have been used during the implementation of the *"Heat Method"*.

Using the heat method for computing the geodesic distances gives some advantages. One advantage, it is highly flexible to be applied on various types of geometric discretization such as point clouds, polygonal surfaces, simplicial mesh, etc. Other advantage, heat method works with sparse linear systems, which can be prefactored once and solved subsequently many times with a notable speed.

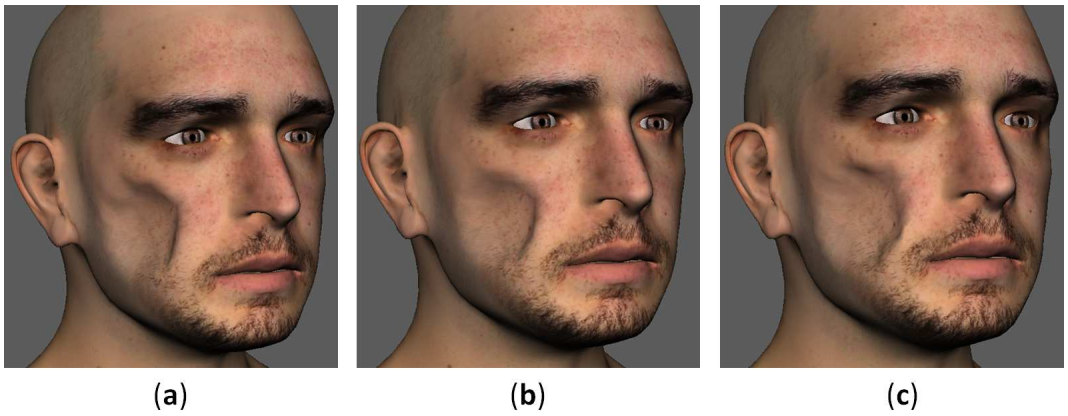

Figure 5.5: The surface responds after collision under different projected constraints. (a) the target surface with stretching constraint. (b) the target surface with stretching and shearing constraints. (c) the target surface with stretching, shearing and bending constraints.

5.5 General Algorithm and Implementation

The proposed framework is specialized to operate on 3D facial models (triangulated mesh), which can be considered as mesh of particles. Each particle (or vertex) on the mesh has been assigned to some physical properties such as mass (m) and position (q). Velocities are omitted because they are already calculated implicitly. The other important data sets are acceleration (a) from the assigned force and time step size (Δt).

Based on the explanations in the previous sections and data sets, the overall procedure, which performs the simulation underneath of our framework can be described by algorithm 1.

Our framework has been implemented as a plugin for Maya 2016 [Aut] using Python. Third party packages such as Scipy, Numpy and Scikits.sparse have been used for matrix operations. During the implementation, it should be noted that the cotangent computation of the heat method shown at the beginning of algorithm 1 is performed independent from the cosine calculations in position constraints in the end of algorithm 1. Another important parameter, which we have not discussed yet, is the stiffness (k_{cons}) of the constraint. According to Müller et al. [MHHR07], a smooth alternative to apply stiffness parameter is to multiply the displacements by k_{cons}. We have followed the same direction and applied the stiffness parameter only to the bending constraint. This straightforward calculation has been performed implicitly inside the bending constraint $\Delta q_{be} \leftarrow \Delta q_{be}(1 - k_{cons})$ so it has not been reflected in algorithm 1 just to prevent confusion.

input : A triangle mesh $M_f = (V, E, F)$ and a selected vertex $v_i \in V$
output: 2 step output: First, geodesic distances $d(v_i, v_j)$. Second, performing the simulation inside the boundaries of the geodesic distances
$assignBoundaries(v_i, Distance_{min}, Distance_{max})$;
foreach *face (F) in* M_f **do**
  foreach *edge (E) in F* **do**
    Compute the cotangent of the angles for Laplacian coordinates and areas of the faces by using: $\cot\alpha \leftarrow \frac{E_1 \cdot E_2}{|E_1 \times E_2|}$;
  end
  $A_i \leftarrow computeFaceArea(\alpha)$;
end
`/* integrate the heat flow */`
$\dot{u} = \Delta u$;
`/* evaluate the normalized gradient field */`
$X = -\frac{\nabla u}{|\nabla u|}$;
`/* solve the Poisson equation */`
$\Delta d = \nabla \cdot X$;
`/* simulate points(or vertices) (q) of the mesh inside the boundaries of the heat area */`
foreach $q \in d(v_i, v_j)$ **do**
  $q_{n+1} \leftarrow q_n + (q_n - q_{n-1})(1-k) + a_n \Delta t^2$;
  foreach $specifiedIterations$ **do**
    $\Delta q_{col} \leftarrow collisionConstraint(q_{n+1})$;
    $q_{n+1} \leftarrow q_{n+1} + \Delta q_{col}$;
    $\Delta q_{st} \leftarrow stretchingConstraint(q_{n+1})$;
    $q_{n+1} \leftarrow q_{n+1} + \Delta q_{st}$;
    $\Delta q_{sh} \leftarrow shearingConstraint(q_{n+1})$;
    $q_{n+1} \leftarrow q_{n+1} + \Delta q_{sh}$;
    $\Delta q_{be} \leftarrow bendingConstraint(q_{n+1})$;
    $q_{n+1} \leftarrow q_{n+1} + \Delta q_{be}$;
  end
  $updateOutputGeometry(q_{n+1})$
end

Algorithm 1: General Algorithm for the Proposed Verlet Framework

Figure 5.6: "Cat Girl" was tested with different damping parameter. (a) $k_{damp} = 0.3$. (b) $k_{damp} = 0.6$. (c) $k_{damp} = 0.9$.

5.6 Validation and Experimental Results

The examples in this section were performed on a 4-core Intel i7-2600 3.4 GHz machine with 8 GB of RAM and an nVidia GTX 570 GPU. We tested and validated the proposed framework with different facial models using single threaded CPU implementation for Autodesk Maya 2016. In our experimental validations, the main purpose was the observation of how the surface responds during collision under different constraint types and to verify the adaptation of the heat method for the developers who already use position-based dynamics.

Figure 5.7 and table 5.1 present the timings to create the soft area on the surface of the mesh. Both complexity of the mesh and the size of the area are the key factors for the run-time. In figure 5.8, the desired size of the area is illustrated in different ranges according to the provided chart in figure 5.7. The values clearly show the benefit of exact area selection with the heat method. Therefore in our experiments, we have kept the selection of the target area according to its natural size. Each region of the face is assigned to different size control parameters according to the proportions of the model structure. For example cheek region is slightly wider than chin but narrower than forehead.

Moreover, the performance of the simulation is strictly dependent on the size of the selected area. Graph from figure 5.9 and table 5.2 summarize the average timings according to the range of the selected area. During measuring these timings, we performed 2 iterations to the projected constraints and handled a single collision

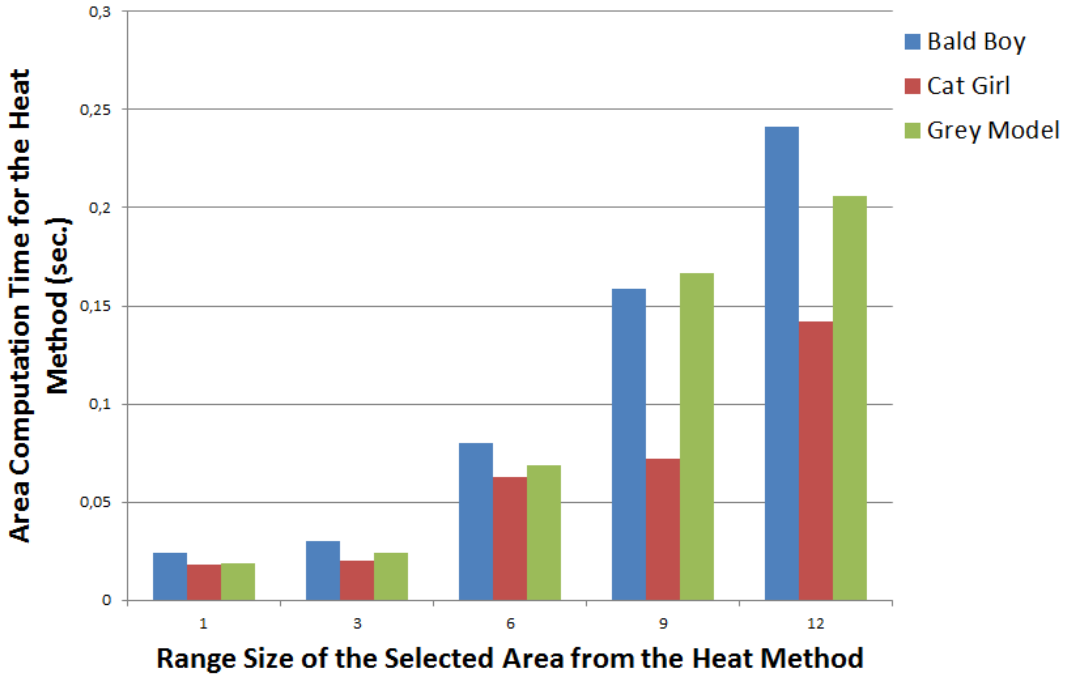

Figure 5.7: Comparison chart of the heat method speed between the example facial models. "Bald Boy" model includes 9562 vertices, "Cat Girl" model includes 6798 vertices and "Grey Model" includes 6720 vertices.

contact for the same scenario to all three models. An example case can be seen in figure 5.1. For our other experiments, we kept the range of the size control parameter for the selected area $distance_{max} \leq 3$ for better performance. The main advantage of the heat method can be observed clearly from the figure 5.9. If the simulation scenario does not require the deformation of the whole mesh, selection of the particular area on the mesh definitely increases the performance of the simulation.

Figure 5.5 shows the surface responds after the collision. Each constraint computation is handled independently but the attachment of these constraints in the simulation loop of algorithm 1 has increased the quality and smoothness of the surface. In figures 5.1 and 5.5 the damping parameter from equation 5.3 was kept as 0. However, in

Model	$\#Vertices$	$\#Edges$	$\#Faces$	$dist$=1	$dist$=3	$dist$=6	$dist$=9	$dist$=12
Bald Boy	9562	28416	18848	0.0239	0.0299	0.0799	0.1589	0.241
Cat Girl	6798	20213	13412	0.018	0.02	0.0629	0.072	0.1419
Grey Model	6720	19984	13264	0.019	0.0239	0.069	0.167	0.2059

Table 5.1: Summary of the processed datasets during the heat method computation. *dist* represents the radius range of the arbitrary selected geodesic circle. The cost increases according to the size arbitrary selected area. All computation times are in seconds.

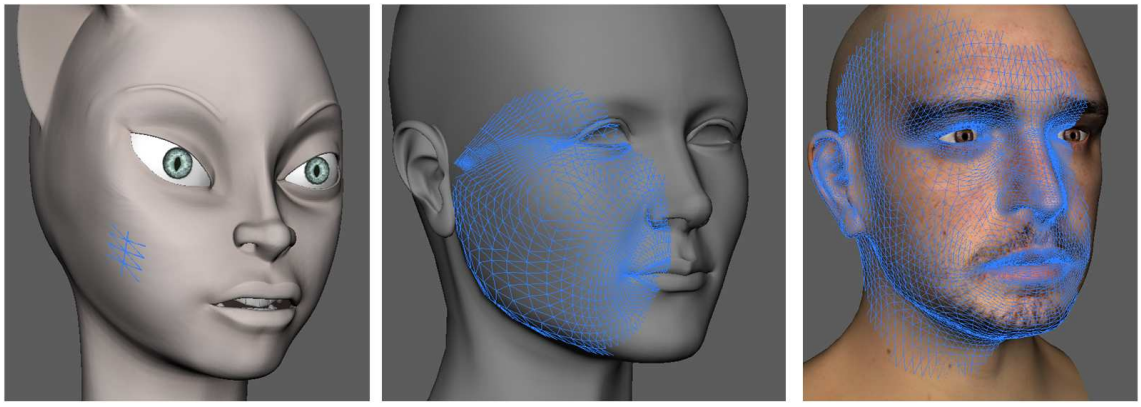

Figure 5.8: Example illustration of the selected area by using the heat method with different ranges. (a) shows the area with 1 unit range on the "Cat Girl" (b) shows the area with 6 units range on the "Grey Model". (c) shows the area with 12 units range on the "Bald Boy"

figure 5.6 the damping parameter varies between [0.3,0.9]. The relationship between damping value and vertex movements is inverse. Assigning the damping parameter to extreme high values causes the selected surface to react like a rigid body.

The proposed framework intrinsically operates over the surface of the mesh independent from the prearranged mechanics, which lie beneath the model. Figure 5.10 demonstrates another case with a blendshape model, which consists of 40 blendshape targets. We created a sample pose, which corresponds to "pain" expression. Our framework performed a collision handling test over the surface of the same model. The existence of the rig elements does not interrupt the computations of the heat method and the position constraints. Figure 5.11 illustrates the whole process of the framework. It is important to note that while the collision object contacts with the model, deformations are limited only inside of the soft area boundaries. Outside of the boundaries, model conserves its rigid behavior.

Model	$\#Vertices$	$\#Edges$	$\#Faces$	$dist$=1	$dist$=3	$dist$=6	$dist$=9	$dist$=12	Whole Model
Bald Boy	9562	28416	18848	0.37	0.9	6.66	15.36	21.22	29.6
Cat Girl	6798	20213	13412	0.24	0.44	3.54	8.52	11.76	18.31
Grey Model	6720	19984	13264	0.25	1.15	6.41	12.44	16.08	19.94

Table 5.2: Summary of the processed datasets for each simulation step. *dist* represents the radius range of the arbitrary selected geodesic circle. The cost increases according to the size arbitrary selected area. All computation times are in seconds.

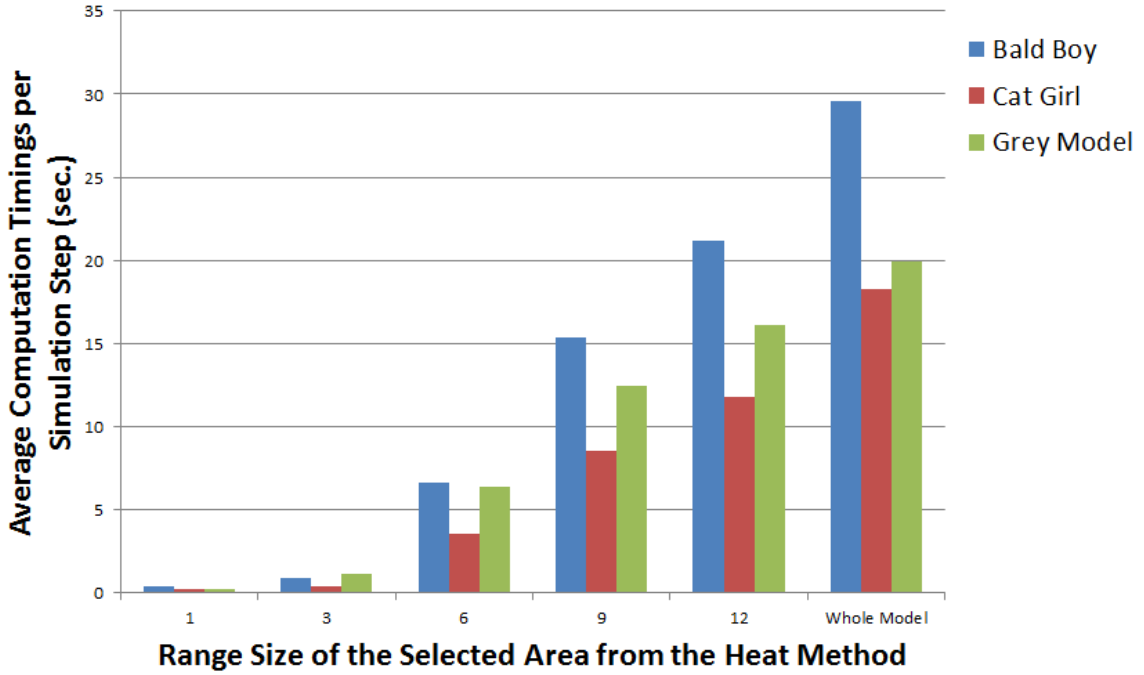

Figure 5.9: Comparison chart of the deformation speeds between the example facial models. "Bald Boy" model includes 9562 vertices, "Cat Girl" model includes 6798 vertices and "Grey Model" includes 6720 vertices.

5.7 Summary

Our Verlet framework allows for the simulation of deformable surfaces on the selected area of the 3D mesh with a desired range. Underneath the framework, we used the recent heat method for the geodesic distance computations and Verlet integration scheme with the position based constraints projections. We have summarized in detail the foundations of position based dynamics and heat method. We have presented its practical usage with different facial models under several conditions. The proposed framework has been specifically designed for the facial models while nevertheless it is particularly suited for the background characters in 3D animation. Furthermore, it is flexible not only to extend it to other application domains easily, but also it can nicely incorporate with the other existing position based dynamics frameworks.

Figure 5.10: "Grey Model" consists of 40 blendshape targets. After creating the pose, the interaction test was performed by using the proposed framework.

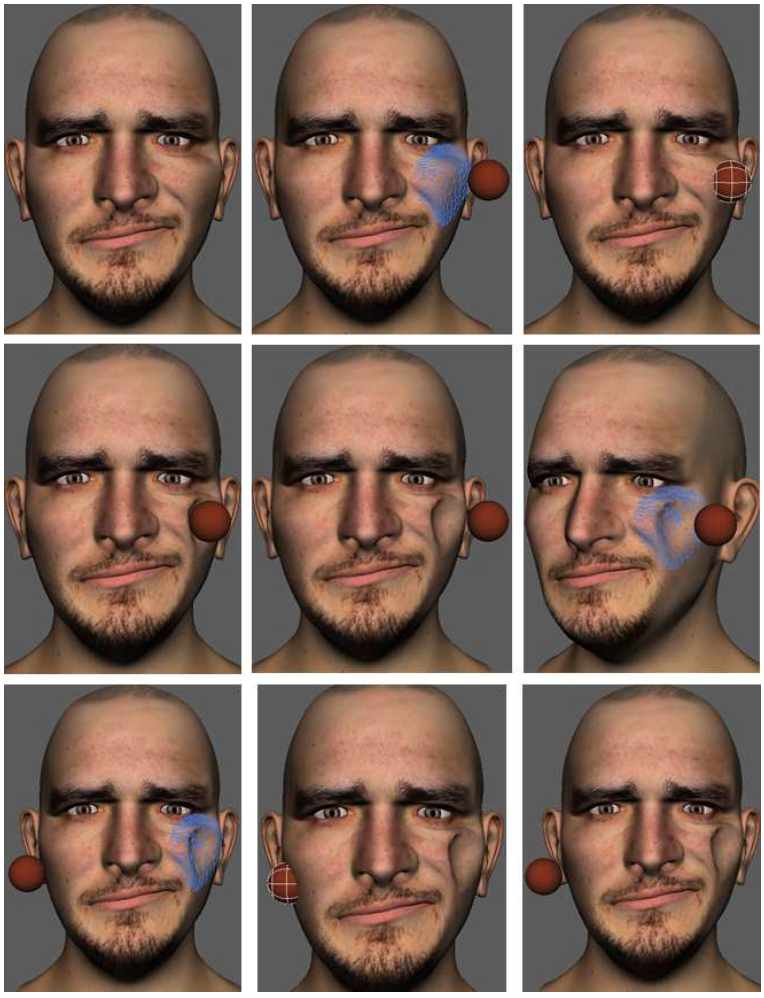

Figure 5.11: An example demonstration of our framework on the "Bald Boy" model, which is posed with blendshapes. The collision object is indicated with the red sphere, which is created by selecting an arbitrary vertex from the model. Immediately after this selection, automatically a soft deformable area occurs in the range of the assigned size control parameter. The area of the soft surface is painted with blue. Since the collision is handled within the boundaries, expected deformation occurs. During the same simulation session, if the red sphere contacts with the model outside of the mentioned boundaries, deformation does not occur.

Chapter 6

Conclusion

In this dissertation, we have proposed three efficient facial animation approaches that are designed to facilitate the artist's pose editing process for creating high-quality facial animation. Our first approach describes the direct manipulation of blendshape models with a sketching interface. We have enhanced the mathematical framework of the original direct manipulation method [LA10] by confining the weight edits to the local geometry of the model. This is done by computing the geodesic circle around the constraint point and limiting the edits to occur only within the circle. Our method's intuitive sketching interface provides the user an easy and interactive manipulation of the relevant blendshapes by using the horizontal curves concept. Our second approach presents the novel transposition based blendshape direct manipulation which eliminates the instability of the pseudo-inverses from the mathematical formulation of the direct manipulation. We, thereby, provide a plain projection of the pin movements to the blendshape deformation directions. Our solution proves that the proposed method keeps the unexpected movements of the face in a minimum level which is almost not observable. Our third approach presents a physics-based deformation framework by combining verlet integration with position based dynamics. Locally constrained deformation is enabled by geodesic distance computation based on heat kernel. The work addressed in this dissertation eases the animation process and can be easily integrated into the existing animation workflows. This chapter concludes this dissertation by summarizing the presented methods and discusses the future directions.

6.1 Conclusion

Producing a detailed facial expression requires a refined character rig and a user interface that is designed to ease the rig manipulation. Blendshape is a predominant technique for high quality facial rigging. In blendshape facial animation, artist defines the morph targets with their corresponding weights and blends them together to obtain a plausible facial poses for each keyframe. Traditionally, the corresponding target weights have been edited directly by the artists, using an interface in which each weight is presented as a slider. Creating all the facial poses of each character by using a slider interface is a slow and manual process and takes a significant human effort for production quality models that may include hundreds of blendshape targets. For example, *Gollum* character had 946 blendshape targets in the *Lord of the Rings* movie [Rai04].

In chapter 3, we present a generic facial sketching direct manipulation approach that provides easy, rapid and interactive editing of blendshape models with simple freehand stroke drawings. Our approach allows the artists to work directly on the surface of the model and keeps them away from the technical details of the approach. The strokes can be drawn directly on the 3D facial model. The method proposed in this dissertation drives the underlying blendshape targets and works with any character style from realistic to cartoon models. The theoretical background of the approach is based on the mathematical framework of direct manipulation blendshape method [LA10]. Although the original framework is a significant advance, the formulation causes a global deformation impact during editing a particular part of the face. We have improved the original formulation and confined the weight edits to the local geometry of the model. This is an important concept as artists prefer to see the edits are applied only in the local geometry. To this end, we have employed the heat method [CWW13] to calculate the geodesic circles around each constraint point and allow the edit of the target weights, which occur only within the circle. Furthermore, we have presented the *horizontal curves* approach for our sketching interface that provides an intuitive editing by using curve entities.

In film and video-game production, artists often create new facial poses from a neutral face model. For each pose, artists should directly modify the geometry by altering the a set of vertices manually or edit the multiple control parameters of the rig elements to obtain a new expression. This process requires individual manipulation of the rig controls in a discontinuous way, which means that each rig element has to be altered one by one. Our solution allows to manipulate several blendshape targets at the same

time via a single curve sketched directly on the model. We allow to create complex deformations on the face with a single and continuous gesture of the hand. Our intuitive user interface tends to reduce the production of facial poses from hours to minutes.

Our approach works with blendshaped polygonal facial models. We performed the implementation of the heat method on the triangulated facial meshes, because of their common usage in production environment. Since our system directly incorporates the blendshape structure of the model, all deformations generated with the curve and manipulators are always constrained by the beneath target shapes. Although the system targets to the experienced artists, the only expected artistic skill is defined as the interface accepts the input stroke inside the boundaries of the 3D model. Therefore, this allows the user to sketch inside the boundaries of the 3D model and create the desired facial poses freely and easily.

In chapter 4, we present a novel and stable transposition based blendshape direct manipulation method which is inspired from the *Jacobian Transpose Method*. During the last decade, the pseudo-inverse based formulations have dominated the direct manipulation frameworks. However, pseudo-inverse is well known with its instability. Therefore, the pseudo-inverse based frameworks cause unstable and unintuitive results during the artistic facial pose editing for keyframe animation. To this end, we present a transposition based mathematical framework for blendshape direct manipulation to enhance the stability of the final results. We provide a simple *pin-and-drag* interface similar to [LA10]. The deformation directions are extracted from the blendshape input matrix and we provide a plain projection of the pin movements onto the deformation directions. The obtained results are more stable and intuitive than the existing pseudo-inverse based mathematical frameworks.

Physically based deformable models have been a dominant subject in the field of computer animation. Physics-based simulation is an interdisciplinary area, which combines Newtonian dynamics, continuum mechanics, kinematics, numerical methods, differential geometry, etc. Deformable solid simulation has been widely employed to generate realistic visual effects of the soft bodies. However, most of the existing frameworks have applied the deformation to the whole model based on mass-spring networks or classic elasticity theory. Besides, applying sophisticated theories to the whole model reduces the efficiency of the computation. To this end, in chapter 5, we present a physics-based framework to simulate surface deformation of the 3D cartoon models. Our framework combines the verlet integration with position based dynamics. Furthermore, we provide an interactive selection of the deformation influence area

with an arbitrary range. Therefore the physical deformation occurs inside the defined boundaries of the desired area of the model. We have performed the geodesic distance of the surface from an arbitrary selected point to a desired range based on heat kernel. Outside of the selected boundaries, we allow the model to conserve its rigid behavior. By following the position-based dynamics method [MHHR07], we have applied the position manipulations by computing the geometric constraints such as stretch, shear and bend to the surface components for collision handling. Our approach is more efficient than applying the physical deformation to the whole model in terms of performance (around 30 times faster by keeping the circle range in *3* units). For the same deformation results on the facial mesh, our solution significantly reduces the cost of computation and provides physically plausible deformation under collision and contact.

In parallel to the position-based approaches, our framework provides the direct position manipulation of the model during the simulation with an implicit velocity updates. Our solution in the framework keeps a smooth balance in terms of simplicity, generality and performance. Furthermore, the framework holds a considerable potential to complement the facial animation systems and it can nicely incorporate the other existing position-based dynamics frameworks such as [BKCW14], [DCB14], [MC11], etc.

6.2 Future Directions

The focus of this dissertation has been reducing the human efforts in the facial animation creation process. However, many challenges still remain open for research and development. We conclude this dissertation by suggesting some potential future research directions.

6.2.1 Blendshape Facial Animation

During blendshape editing, animators are limited with the deformation space of the blendshape targets. An interactive method could be useful to detect the existing target shapes, and automatically produce new target shapes arbitrarily to the desired area of the model to increase the smoothness of the poses without violating the existing blendshape targets. Further, another direction can be considered as an automatic interaction model between the blendshape targets and articulated joints. Therefore,

both rig elements can be used together as a unified rig structure and this coupling increases the visual quality of the poses and expressiveness. Besides, the interpolation of the target shapes composes the desired expressions with a rigid transition. However, this rigidity is not fully natural according to the artist's perspective. Recent research from Barrielle et al. [BSC16] addressed this problem by combining blendshapes with physics-based techniques to obtain almost life-like facial expressions. Further developments are still required to gain fully natural facial expression effects.

6.2.2 Sketch-Based Interfaces

Using strokes and curves is a natural and continuous way to obtain plausible deformations for models. The application domain of sketch based interfaces is still rather limited in animation pipelines. One possible future direction can be adding the time as a constraint to the sketch based frameworks, which provides interactive editing during the execution of the animation. Guay et al. [GRGC15b] explored the space-time curve to animate the body skeleton with smooth transitions between poses. It would be interesting to transfer the concept of *space-time curve* from the whole body skeleton animation to facial animation. Another extension can be considered as creating keyframe poses for hands and feet with the sketched curves. Controlling the crowd simulations with sketches would be a significant development for future by mapping the individual character motions to the crowd with the drawn strokes.

6.2.3 Physics for Face Animation

Combining the traditional rig elements with physical laws is a required step to obtain fully natural facial expressions with secondary effects. Hahn et al. [HTC+13] offered to combine rig parameters with a physically-based volumetric skinning method. In parallel to their approach, it would be interesting to directly couple the rig controls with position-based dynamics approach. New geometric constraints are required to define the elastic and visco-elastic behavior of the skin tissue by considering the skin aging. The facial hair such as beard, eyebrow or eyelash remains open for research. The existing techniques for hair can be extended to the facial hair. Finally, we would like to explore new techniques on geodesic distance computations, which fit to only character face models for calculating more accurate distance measures on facial regions. This would be a significant complement for facial physical simulations to gain realistic results.

Appendix A

Derivation of Geometric Constraints

The constraint function for **stretching** from equation 5.8 is $C_1(q_{i,j}) = |D| - l_{ij}$ where $D = |q_j - q_i|$. The corresponding gradients are $\nabla_{q_i} C = -n$ and $\nabla_{q_j} C = n$ where $n = \frac{q_j - q_i}{|q_j - q_i|}$. After substituting the gradients to equation 5.7, the Lagrange multiplier becomes $\lambda = \frac{|q_j - q_i| - l}{|w_j + w_i|}$ where $w_i = w_j = 1$. and final corrections are [MHHR07]:

$$\Delta q_i = -\frac{1}{2}(|q_j - q_i| - l)(\frac{q_j - q_i}{|q_j - q_i|}) \tag{A.1}$$

$$\Delta q_j = +\frac{1}{2}(|q_j - q_i| - l)(\frac{q_j - q_i}{|q_j - q_i|}) \tag{A.2}$$

The constraint function for **shearing** from equation 5.9 is defined as $C(q_{i,j,k}) = cos^{-1}(D) - \gamma_{ijk}$, where $D = M_{ij} \cdot M_{ik}$ from equation 5.11. With $\frac{d}{dx} cos^{-1}(x) = -\frac{1}{\sqrt{1-x^2}}$, the corresponding gradients are obtained as follows:

$$\nabla_{q_i} C = -\frac{1}{\sqrt{1 - D^2}}(\frac{\partial M_{ij}}{\partial q_i} \cdot M_{ik} + M_{ij} \cdot \frac{\partial M_{ik}}{\partial q_i}) \tag{A.3}$$

$$\nabla_{q_j} C = -\frac{1}{\sqrt{1 - D^2}}(\frac{\partial M_{ij}}{\partial q_j} \cdot M_{ik} + M_{ij} \cdot \frac{\partial M_{ik}}{\partial q_j}) \tag{A.4}$$

$$\nabla_{q_k} C = -\frac{1}{\sqrt{1 - D^2}}(\frac{\partial M_{ij}}{\partial q_k} \cdot M_{ik} + M_{ij} \cdot \frac{\partial M_{ik}}{\partial q_k}) \tag{A.5}$$

After substituting the gradients to equation 5.7, the Lagrange multiplier becomes:

$$\lambda = -\frac{cos^{-1}(M_{ij} \cdot M_{ik}) - \gamma_{ijk}}{|\nabla_{q_i} C|^2 + |\nabla_{q_j} C|^2 + |\nabla_{q_k} C|^2} \tag{A.6}$$

where $w_i = w_j = w_k = 1$. The corrections can be computed easily for the shearing constraint by substituting the gradients and Lagrange multiplier in equation 5.6. We use a middle step to demonstrate the calculations:

$$r_1 = -r_2 - r_3 \tag{A.7}$$

$$r_2 = -\frac{q_i - q_k}{|q_j - q_i||q_k - q_i|} \tag{A.8}$$

$$r_3 = -\frac{q_i - q_j}{|q_j - q_i||q_k - q_i|} \tag{A.9}$$

The general final correction for shearing constraint is:

$$\Delta q_i = -\frac{(\sqrt{1 - D^2})(cos^{-1}(D) - \gamma_{ijk})}{|r_1|^2 + |r_2|^2 + |r_3|^2} r_1 \tag{A.10}$$

$$\Delta q_j = -\frac{(\sqrt{1 - D^2})(cos^{-1}(D) - \gamma_{ijk})}{|r_1|^2 + |r_2|^2 + |r_3|^2} r_2 \tag{A.11}$$

$$\Delta q_k = -\frac{(\sqrt{1 - D^2}(cos^{-1}(D) - \gamma_{ijk}))}{|r_1|^2 + |r_2|^2 + |r_3|^2} r_3 \tag{A.12}$$

The constraint function for **bending** from equation 5.10 is defined as $C(q_{i,j,k,l}) = cos^{-1}(D) - \theta_{ijkl}$, where $D = N_{ijk} \cdot N_{ijl}$ from equation 5.12. According to [MHHR07] q_i is set to 0 ($q_i = 0$), with $\frac{d}{dx} cos^{-1}(x) = -\frac{1}{\sqrt{1-x^2}}$, the corresponding gradients are obtained as follows:

$$\nabla_{q_i} C = -\nabla_{q_j} C - \nabla_{q_k} C - \nabla_{q_l} C \tag{A.13}$$

$$\nabla_{q_j} C = -\frac{1}{\sqrt{1 - D^2}} ((\frac{\partial N_{ijk}}{\partial q_j})^T N_{ijl} + (\frac{\partial N_{ijl}}{\partial q_j})^T N_{ijk}) \tag{A.14}$$

$$\nabla_{q_k} C = -\frac{1}{\sqrt{1-D^2}}((\frac{\partial N_{ijk}}{\partial q_k})^T N_{ijl}) \tag{A.15}$$

$$\nabla_{q_l} C = -\frac{1}{\sqrt{1-D^2}}((\frac{\partial N_{ijl}}{\partial q_l})^T N_{ijk}) \tag{A.16}$$

After substituting the gradients to equation 5.7, the Lagrange multiplier becomes:

$$\lambda = -\frac{cos^{-1}(N_{ijk} \cdot N_{ijl}) - \theta_{ijkl}}{|\nabla_{q_i} C|^2 + |\nabla_{q_j} C|^2 + |\nabla_{q_k} C|^2 + |\nabla_{q_l} C|^2} \tag{A.17}$$

where $w_i = w_j = w_k = w_l = 1$. By following the form mentioned in [MHHR07], as a middle step, we take advantage of the following computations before finding the final corrections:

$$r_1 = -r_2 - r_3 - r_4 \tag{A.18}$$

$$r_2 = -\frac{q_k \times N_{ijl} + (N_{ijk} \times q_k)D}{|q_j \times q_k|} - \frac{q_l \times N_{ijk} + (N_{ijl} \times q_l)D}{|q_j \times q_l|} \tag{A.19}$$

$$r_3 = \frac{q_j \times N_{ijl} + (N_{ijk} \times q_j)D}{|q_j \times q_k|} \tag{A.20}$$

$$r_4 = \frac{q_j \times N_{ijk} + (N_{ijl} \times q_j)D}{|q_j \times q_l|} \tag{A.21}$$

The general final correction for bending constraint is:

$$\Delta q_i = -\frac{(\sqrt{1-D^2})(cos^{-1}(D) - \theta_{ijkl})}{|r_1|^2 + |r_2|^2 + |r_3|^2 + |r_4|^2} r_1 \tag{A.22}$$

$$\Delta q_j = -\frac{(\sqrt{1-D^2})(cos^{-1}(D) - \theta_{ijkl})}{|r_1|^2 + |r_2|^2 + |r_3|^2 + |r_4|^2} r_2 \tag{A.23}$$

$$\Delta q_k = -\frac{(\sqrt{1-D^2}(cos^{-1}(D) - \theta_{ijkl}))}{|r_1|^2 + |r_2|^2 + |r_3|^2 + |r_4|^2} r_3 \tag{A.24}$$

$$\Delta q_l = -\frac{(\sqrt{1-D^2})(cos^{-1}(D) - \theta_{ijkl})}{|r_1|^2 + |r_2|^2 + |r_3|^2 + |r_4|^2} r_4 \tag{A.25}$$

Appendix B

Overview of Heat Method

The heat method of [CWW13] can be applied to wide variety of model surfaces like triangulated meshes, quadrangulated meshes, point clouds, etc. In this thesis, the heat method is applied on the triangulated models. However, adapting the heat method on other surfaces is just an implementation issue which will be undertaken in future.

For the triangular meshes, the Laplacian discretization [Sor06] for an arbitrary vertex "i" is defined as:

$$(Lu)_i = \frac{1}{2A_i} \sum_j (\cot \alpha_{ij} + \cot \beta_{ij})(u_j - u_i) \tag{B.1}$$

where A_i is the area of the Voronoi cell of i, α_{ij} and β_{ij} are the angles of the opposite edges. As the second step of the heat method, the vector field should be evaluated. Before evaluating the vector field, the gradients of the given triangle is computed as:

$$\nabla u = \frac{1}{2A_f} \sum_i u_i (N \times e_i) \tag{B.2}$$

where A_f is the primitive face area, N is the unit normal corresponding to the face and e_i is the edge vector. After, the vertex "i" integrated divergence is computed as:

$$\nabla \cdot X = \frac{1}{2} \sum_j \cot \theta_1 (e_1 \cdot X_j) + \cot \theta_2 (e_2 \cdot X_j) \tag{B.3}$$

where the sum is computed over primitive triangles j with X_j vector, e_1 and e_2 are the edge vectors of the corresponding triangle j which has the i, θ_1 and θ_2 are the

opposite angles. If $b \in R^{|V|}$ is the vector divergences of X, after as the last step, the final geodesic distance is computed by solving the Poisson problem:

$$L_c \mathbf{d} = b \tag{B.4}$$

where L_c is the cotan operator for the remaining sum and $\mathbf{d}$ is the function of geodesic distance approximations. For further details on the heat method, we refer the reader to [CWW13].

Appendix C

Transition from Pseudo-Inverse to Transposition

First, we orthonormalize the B^T matrix and extract the deformation directions from the blendshape matrix because both share the same vector spans. Therefore, we apply the *Gram-Schmidt* process and store the extracted orthonormal matrix in Q. After, we substitute the new Q matrix instead of B^T in equation 4.2 with a scaling step-size matrix R for the point movements. The new equation becomes:

$$w = (QQ^T)^{-1}Q(Rm) \tag{C.1}$$

In equation C.1, Q is an orthonormal matrix. Thereby, QQ^T becomes an *identity* matrix. The new update function turns into $w = Q(Rm)$. According to Jacobian Transpose suggestion [Wel93], weight update can be represented as $w = B^T m$. To minimize the scaling step-size of the point movements we apply the following:

$$\min_R \|B^T - QR\|^2 \tag{C.2}$$

The solution of equation C.2 shows us $R = Q^T B^T$. Alternatively, the least square process can be replaced with a simple *qr factorization*. Then, R is substituted to equation C.1 and the weight update function becomes:

$$w = B^T m \tag{C.3}$$

References

[AAJ03] B. R. D. Araújo, J. Armando, and P. Jorge. Blobmaker: Free-form modelling with variational implicit surfaces. In *In Proc. of the 12th Portuguese Computer Graphics Meeting*, pages 17–26, 2003.

[ACP03] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.

[AGB04] A. Alexe, V. Gaildrat, and L. Barthe. Interactive modelling from sketches using spherical implicit functions. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '04, pages 25–34, New York, NY, USA, 2004. ACM.

[ARV07] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[ASK+05] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 408–416, New York, NY, USA, 2005. ACM.

[ASSJ06] F. Anastacio, M. C. Sousa, F. Samavati, and J. A. Jorge. Modeling plant structures using concept sketches. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '06, pages 105–113, New York, NY, USA, 2006. ACM.

[ATL12] K. Anjyo, H. Todo, and J. P. Lewis. A practical approach to direct manipulation blendshapes. *Journal of Graphics Tools*, 16(3):160–176, 2012.

[Aut] Inc. Autodesk. Maya 2016.

[AZ10] O. Aina and J. Zhang. Automatic muscle generation for physically-based facial animation. In *Proc. SIGGRAPH*, 2010.

[BBPV03] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–650, 2003.

[BETC12] J. Bender, K. Erleben, J. Trinkle, and E. Coumans. Interactive simulation of rigid body dynamics in computer graphics. In *EUROGRAPHICS 2012 State of the Art Reports*, 2012.

[BGY+13] K. S. Bhat, R. Goldenthal, Y. Ye, R. Mallet, and M. Koperwas. High fidelity facial animation capture and retargeting with contours. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 7–14, New York, NY, USA, 2013. ACM.

[BHB+11] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. In *Proc. SIGGRAPH Papers*, 2011.

[BKCW14] J. Bender, D. Koschier, P. Charrier, and D. Weber. Position-based simulation of continuous materials. *Computers & Graphics*, 44(0):1 – 10, 2014.

[BL85] P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements in the computer-generated animated short 'tony de peltrie'. In *Proc. SIGGRAPH Papers*, 1985.

[BLB+08] B. Bickel, M. Lang, M. Botsch, M. A. Otaduy, and M. Gross. Pose-space animation and transfer of facial details. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 57–66, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[BMF03] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003.

[BMM15] J. Bender, M. Müller, and M. Macklin. Position-based simulation methods in computer graphics. In *EUROGRAPHICS 2015 Tutorials*, 2015.

[BMOT13] J Bender, M Müller, M. A. Otaduy, and M. Teschner. Position-based methods for the simulation of solid objects in computer graphics. In *EUROGRAPHICS 2013 State of the Art Reports*, 2013.

[BMWG07] M. Bergou, S. Mathur, M. Wardetzky, and E. Grinspun. Tracks: Toward directable thin shells. *ACM Trans. Graph.*, 26(3), 2007.

[BMZB02] H. Biermann, I. M. Martin, D. Zorin, and F. Bernardini. Sharp features on multiresolution subdivision surfaces. *Graphical Models*, 64(2):61–77, 2002.

[BOP98] S Basu, N. Oliver, and A. Pentland. 3d modeling of human lip motion. In *ICCV'98*, pages 337–343, 1998.

[BP81] N. I. Badler and S. M. Platt. Animating facial expressions. In *Proc. SIGGRAPH Papers*, 1981.

[BP07] I. Baran and J. Popovic. Automatic rigging and animation of 3d characters. In *Proc. SIGGRAPH Transactions on Graphics*, 2007.

[BSC16] Vi. Barrielle, N. Stoiber, and C. Cagniart. Blendforces: a dynamic framework for facial animation. *Computer Graphics Forum (Proceedings of Eurographics 2016)*, 2016.

[BV99] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998.

[BWD13] J. Bender, D. Weber, and R. Diziol. Fast and stable cloth simulation based on multi-resolution shape matching. *Computers & Graphics*, 37(8):945 – 954, 2013.

[CBE+15] M. Cong, M. Bao, Jane L. E, K. S. Bhat, and R. Fedkiw. Fully automatic generation of anatomical face simulation models. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '15, pages 175–183, New York, NY, USA, 2015. ACM.

[CDB02] E. S. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *In Proceedings of Pacific Graphics*, pages 68–76, 2002.

[CHP89] J.E. Chadwick, D.R. Haumann, and R.E. Parent. Layered construction for deformable animated characters. In *ACM Siggraph Computer Graphics*, volume 23, pages 243–252. ACM, 1989.

[CJ08] E. Chang and O. Jenkins. Sketching articulation and pose for facial animation. In *Data-Driven 3D Facial Animation*, pages 145–161. Springer London, 2008.

[CK05a] B. Choe and H.S. Ko. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[CK05b] M. G. Choi and H.S. Ko. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):91–101, 2005.

[CLK01] B. Choe, H. Lee, and H.S. Ko. Performance-driven muscle-based facial animation. *Journal of Visualization and Computer Animation*, 12(2):67–79, 2001.

[CNJC03] M. Contero, F. Naya, J. Jorge, and J. Conesa. Cigro: A minimal instruction set calligraphic interface for sketch-based modeling. In *Proceedings of the 2003 International Conference on Computational Science and Its Applications: PartIII*, ICCSA'03, pages 549–558, Berlin, Heidelberg, 2003. Springer-Verlag.

[COL15] O. Cetinaslan, V. Orvalho, and J.P. Lewis. Sketch-based controllers for blendshape facial animation. In *Eurographics 2015 - Short Papers*, 2015.

[CON05] B.Y. Chen, Y. Ono, and T. Nishita. Character animation creation using hand-drawn sketches. *The Visual Computer*, 21(8-10):551–558, 2005. Pacific Graphics 2005 Conference Proceedings.

[Coq90] S. Coquillart. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In *ACM SIGGRAPH Computer Graphics*, volume 24, pages 187–196. ACM, 1990.

[CS07] F. Cordier and H. Seo. Free-form sketching of self-occluding objects. *IEEE Comput. Graph. Appl.*, 27(1):50–59, January 2007.

[CSSJ05] J. J. Cherlin, F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling with few strokes. In *Proceedings of the 21st Spring Conference on Computer Graphics*, SCCG '05, pages 137–145, New York, NY, USA, 2005. ACM.

[CWLZ13] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41:1–41:10, July 2013.

[CWW13] K. Crane, C. Weischedel, and M. Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph.*, 32(5):152:1–152:11, 2013.

[CXH03] J.X. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 193–206, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[DAC+03] J. Davis, M. Agrawala, E. Chuang, Z. Popović, and D. Salesin. A sketching interface for articulated figure animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 320–328, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[DB96] N. Dubreuil and D. Bechmann. Facial animation. In *Computer Animation'96. Proceedings*, pages 98–109. IEEE, 1996.

[DB00] Van Kreveld M. Overmars M. Schwarzkopf O. De Berg, M. *Computational Geometry: Algorithms and Applications*. Springer, 2000.

[DB13] C. Deul and J. Bender. Physically-based character skinning. In *Virtual Reality Interactions and Physical Simulations (VRIPhys)*, 2013.

[DBB11] R. Diziol, J. Bender, and D. Bayer. Robust real-time deformation of incompressible surface meshes. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 237–246, 2011.

[DCB14] C. Deul, P. Charrier, and J. Bender. Position-based rigid body dynamics. *Computer Animation and Virtual Worlds*, 2014.

[DCFN06] Z. Deng, P.Y. Chiang, P. Fox, and U. Neumann. Animating blendshape faces by cross-mapping motion capture data. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, pages 43–48, New York, NY, USA, 2006. ACM.

[D.D00] Hoffman D.D. *Visual Intelligence: How We Create What We See*. W. W. Norton & Company, 2000.

[DDGG05] K. Das, P. Diaz-Gutierrez, and M. Gopi. Sketching Free-form Surfaces Using Network of Curves. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2005.

[DE03] G.M. Draper and P.K. Egbert. A gestural interface to free-form deformation. In *IN GRAPHICS INTERFACE 2003*, pages 113–120, 2003.

[DJW+06] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M.P. Cani. Virtual Garments: A Fully Geometric Approach for Clothing Design. *Computer Graphics Forum*, 2006.

[DSB99] M. Desbrun, P. Schröder, and A. Barr. Interactive animation of structured deformable objects. In *Proceedings of the 1999 Conference on Graphics Interface '99*, 1999.

[DSP06] K. G. Der, R. W. Sumner, and J. Popović. Inverse kinematics for reduced deformable models. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1174–1179, New York, NY, USA, 2006. ACM.

[EBDP96] I. Essa, S. Basu, T. Darrell, and A. Pentland. Modeling, tracking and interactive animation of faces and heads//using input from video. In *Computer Animation'96. Proceedings*, pages 68–79. IEEE, 1996.

[EHBE97] L. Eggli, C.Y. Hsu, B.D. Bruderlin, and G. Elber. Inferring 3d models from freehand sketches and constraints. *Computer-Aided Design In Solid Modelling*, 29(2):101–112, 1997.

[EP97] I.A. Essa and A.P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):757–763, 1997.

[ESA07] M. Eitz, O. Sorkine, and M. Alexa. Sketch based image deformation. In *Proceedings of Vision, Modeling and Visualization*, pages 135–142, 2007.

[Fau98] F. Faure. Interactive solid animation using linearized displacement constraints. In *9 th Eurographics Workshop on Computer Animation and Simulation*, 1998.

[FBSS04] T. Fleisch, G. Brunetti, P. Santos, and A. Stork. Stroke-input methods for immersive styling environments. In *SMI*, pages 275–283. IEEE Computer Society, 2004.

[FMK+03] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3d models. *ACM Trans. Graph.*, 22(1):83–105, January 2003.

[FP15] M. Fratarcangeli and F. Pellacini. Scalable partitioning for parallel position based dynamics. *Computer Graphics Forum*, 2015.

[FRSS04] T. Fleisch, F. Rechel, P. Santos, and A. Stork. Constraint Stroke-Based Oversketching for 3D Curves. In *Sketch Based Interfaces and Modeling*. The Eurographics Association, 2004.

[GCR13] M. Guay, M.-P. Cani, and R. Ronfard. The line of action: An intuitive interface for expressive character posing. *ACM Trans. Graph.*, 32(6):205:1–205:08, 2013.

[GHDS03] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003.

[GHF+07] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. *ACM Trans. Graph.*, 26(3), 2007.

[GM10] O. Gunnarsson and S. C. Maddock. Sketch-based posing of 3d faces for facial animation. In *Theory and Practice of Computer Graphics*, 2010.

[GMS14] M. Gao, N. Mitchell, and E. Sifakis. Steklov-poincarÉ skinning. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '14, pages 139–148, Aire-la-Ville, Switzerland, Switzerland, 2014. Eurographics Association.

[GRGC15a] M. Guay, R. Ronfard, M. Gleicher, and M.-P. Cani. Adding dynamics to sketch-based character animations. In *Proceedings of the Workshop on Sketch-Based Interfaces and Modeling*, 2015.

[GRGC15b] M. Guay, R. Ronfard, M. Gleicher, and M.-P. Cani. Space-time sketching of character animation. *ACM Trans. Graph.*, 34(4):118:1–118:10, 2015.

[HACH+04] V. Hayward, O.R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-de-la Torre. Haptic interfaces and devices, 2004.

[HCTW11] H. Huang, J. Chai, X. Tong, and H. Wu. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. In *Proc. SIGGRAPH Papers*, 2011.

[HJCW06] M. Hong, S. Jung, M.H. Choi, and S. W. J. Welch. Fast volume preservation for a mass-spring system. *IEEE Computer Graphics and Applications*, 26(5):83–91, 2006.

[HMC+15] F. Hahn, F. Mutzel, S. Coros, B. Thomaszewski, M. Nitti, M. Gross, and R. W. Sumner. Sketch abstractions for character posing. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2015.

[HMT+12] F. Hahn, S. Martin, B. Thomaszewski, R. Sumner, S. Coros, and M. Gross. Rig-space physics. *ACM Trans. Graph.*, 31(4):72:1–72:8, July 2012.

[HSK15] D. Holden, J. Saito, and T. Komura. Learning an inverse rig mapping for character animation. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2015.

[HTC+13] F. Hahn, B. Thomaszewski, S. Coros, R. W. Sumner, and M. Gross. Efficient simulation of secondary motion in rig-space. In *Proceedings*

of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '13, pages 165–171, New York, NY, USA, 2013. ACM.

[IH01] T. Igarashi and J. F. Hughes. A suggestive interface for 3d drawing. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 173–181, New York, NY, USA, 2001. ACM.

[IMKT97] T. Igarashi, S. Matsuoka, S. Kawachiya, and H. Tanaka. Interactive beautification: A technique for rapid geometric design. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, UIST '97, pages 105–114, New York, NY, USA, 1997. ACM.

[IMT99] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[ISF07] G. Irving, C. Schroeder, and R. Fedkiw. Volume conserving finite element simulations of deformable models. *ACM Trans. Graph.*, 26(3), 2007.

[ITF04] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Symposium on Computer Animation*, 2004.

[Jak01] T. Jakobsen. Advanced character physics. In *In Proceedings of the Game Developers Conference*, pages 383–401, 2001.

[JLCW06] Z. Ji, L. Liu, Z. Chen, and G. Wang. Easy mesh cutting. *Computer Graphics Forum*, pages 283–291, 2006.

[JP99] D. L. James and D. K. Pai. Artdefo: Accurate real time deformable objects. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999.

[JT05] D. L. James and C. D. Twigg. Skinning mesh animations. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 399–407, New York, NY, USA, 2005. ACM.

[JTDP03] P. Joshi, W.C. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 187–192, 2003.

[KCMF12] T.Y. Kim, N. Chentanez, and M. Müller-Fischer. Long range attachments - a method to simulate inextensible clothing in computer games. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 305–310, 2012.

[KCZO07] L. Kavan, S. Collins, J. Zara, and C. O'Sullivan. Skinning with dual quaternions. In *2007 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 39–46. ACM Press, April/May 2007.

[KD81] Y. Kurozumi and W.A. Davis. Polygonal approximation by the minimax method. pages 248–264, 1981.

[KDS06] L. B. Kara, C. M. D'Eramo, and K. Shimada. Pen-based styling design of 3d geometry using concept sketches and template models. In *Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling*, SPM '06, pages 149–160, New York, NY, USA, 2006. ACM.

[KGG02] R. Kasturi, L.O. Gorman, and V. Govindaraju. Document image analysis: A primer. *Sadhana*, 27(1):3–22, 2002.

[KH06] O. A. Karpenko and J. F. Hughes. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Trans. Graph.*, 25(3):589–598, July 2006.

[KHS01] K. Kahler, J. Haber, and H.P. Seidel. Geometry-based muscle modeling for facial animation. In *Graphics Interface*, pages 37–46, 2001.

[KHYS02] K. Kahler, J. Haber, H. Yamauchi, and H.P. Seidel. Head shop: generating animated head models with anatomical structure. 2002.

[KJ11] T. Kim and D. L. James. Physics-based character skinning using multi-domain subspace deformations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 63–72, New York, NY, USA, 2011. ACM.

[KJP02] P. G. Kry, D. L. James, and D. K. Pai. Eigenskin: Real time large deformation character skinning in hardware. In *Proceedings of the 2002*

ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '02, pages 153–159, New York, NY, USA, 2002. ACM.

[KMMTT92] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. Simulation of Facial Muscle Actions Based on Rational Free Form Deformations. *Computer Graphics Forum*, 11(3):59–69, May 1992.

[KNE10] M. Kelager, S. Niebe, and K. Erleben. A triangle bending constraint model for position-based dynamics. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2010)*, 2010.

[Koe98] H. Koenig. *Modern Computational Methods*. Taylor & Francis, 1998.

[Kom88] K. Komatsu. Human skin model capable of natural shape variation. In *The Visual Computer*, 1988.

[KPGF07] B. Kubiak, N. Pietroni, F. Ganovelli, and M. Fratarcangeli. A robust method for real-time thread simulation. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology*, pages 85–88, 2007.

[KS05] L. B. Kara and T. F. Stahovich. An image-based, trainable symbol recognizer for hand-drawn sketches. *Comput. Graph.*, 29(4):501–517, August 2005.

[KS06] L. B. Kara and K. Shimada. Construction and Modification of 3D Geometry Using a Sketch-based Interface. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2006.

[LA10] J. P. Lewis and K. Anjyo. Direct manipulation blendshapes. *IEEE Comput. Graph. Appl.*, 30(4):42–50, July 2010.

[LAR+14] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng. Practice and theory of blendshape facial models. In *Eurographics STAR 2014*, pages 199–218, 2014.

[LCA05] C. Larboulette, M.-P. Cani, and B. Arnaldi. Dynamic skinning: Adding real-time dynamic effects to an existing character animation. In *Proc. of the 21st Spring Conference on Computer Graphics*, 2005.

[LCF00] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000.

[LCXS09] M. Lau, J. Chai, Y.Q. Xu, and H.Y. Shum. Face poser: Interactive modeling of 3d facial expressions using facial priors. *ACM Trans. Graph.*, 29(1):3:1–3:17, 2009.

[LD08] Q. Li and Z. Deng. Facial motion capture editing by automated orthogonal blendshape construction and weight propagation. *IEEE Computer Graphics and Applications*, 2008.

[LD10] J. P. Lewis and N. Dragosavac. Stable and efficient differential inverse kinematics. In *ACM SIGGRAPH ASIA 2010 Sketches*, SA '10, pages 1–2, 2010.

[LDSS99] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 343–350, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[Lev07] D. Levinson. Lecture notes in introduction to transportation engineering. University Lectures in University of Minnesota, November 2007.

[LG06] Y. Luo and M.L. Gavrilova. 3d facial model synthesis using voronoi approach. In *Voronoi Diagrams in Science and Engineering, 2006. ISVD'06. 3rd International Symposium on*, pages 132–137. IEEE, 2006.

[LTW95] Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proc. SIGGRAPH '95*, 1995.

[LYWG13] L. Liu, K. Yin, B. Wang, and B. Guo. Simulation and control of skeleton-driven soft body characters. *ACM Trans. Graph.*, 32(6):215:1–215:8, November 2013.

[M.08] Matthias M. Hierarchical position based dynamics. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2008)*, 2008.

[MA07] M. Meyer and J. Anderson. Key point subspace acceleration and soft caching. *ACM Trans. Graph.*, 26(3), July 2007.

[MAO+11] J. C. Miranda, X. Alvarez, J. Orvalho, D. Gutierrez, A. A. Sousa, and V. Orvalho. Sketch express: facial expressions made easy. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pages 87–94, 2011.

[MAO+12] J. C. Miranda, X. Alvarez, J. Orvalho, D. Gutierrez, A. A. Sousa, and V. Orvalho. Sketch express: A sketching interface for facial animation. *Computers & Graphics*, 36(6):585–595, 2012.

[MC10] M. Müller and N. Chentanez. Wrinkle meshes. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 85–92, 2010.

[MC11] M. Müller and N. Chentanez. Solid simulation with oriented particles. *ACM Trans. Graph.*, 30(4), 2011.

[MCC11a] T. McLaughlin, L. Cutler, and D. Coleman. Character rigging, deformations, and simulations in film and game production. In *Proc. SIGGRAPH '11 Courses*, 2011.

[MCC11b] T. McLaughlin, L. Cutler, and D. Coleman. Character rigging, deformations, and simulations in film and game production. In *ACM SIGGRAPH 2011 Courses*, SIGGRAPH '11, pages 5:1–5:18, New York, NY, USA, 2011. ACM.

[McK06] M. McKinley. *The Game Animator's Guide to Maya.* John Wiley & Sons, 2006.

[MCKM14] M. Müller, N. Chentanez, T.Y. Kim, and M. Macklin. Strain based dynamics. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, 2014.

[MG03] A. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 562–568, New York, NY, USA, 2003. ACM.

[MG04] M. Müller and M. Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, 2004.

[MHHR07] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[MHTG05] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, 2005.

[MJC+08] W.C. Ma, A. Jones, J.Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec. Facial performance synthesis using deformation-driven polynomial displacement maps. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 121:1–121:10, New York, NY, USA, 2008. ACM.

[ML07] M. Masry and H. Lipson. A sketch-based interface for iterative design and analysis of 3d objects. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[MS07] T. McLaughlin and S. Sumida. The morphology of digital creatures. In *Proc. of SIGGRAPH '07 Courses*, 2007.

[MSJT08] M. Müller, J. Stam, D. James, and N. Thürey. Real time physics: Class notes. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08, pages 88:1–88:90, 2008.

[MTPT88a] N. Magnenat-Thalmann, E. Primeau, and D. Thalmann. Abstract muscle action procedures for human face animation. *The Visual Computer*, 3(5):290–297, March 1988.

[MTPT88b] N. Magnenat-Thalmann, E. Primeau, and D. Thalmann. Abstract muscle action procedures for human face animation. *The Visual Computer*, 3(5):290–297, March 1988.

[MZS+11] A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.*, 30(4):37:1–37:12, July 2011.

[NF06] G. Nataneli and P. Faloutsos. Sketch-based facial animation. Presented as the 2006 Eurographics / ACM SIGGRAPH Symposium on Computer Animation, 2006.

[NFN00] J. Noh, D. Fidaleo, and U. Neumann. Animated deformations with radial basis functions. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 166–174. ACM, 2000.

[NISA07] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: Designing freeform surfaces with 3d curves. *ACM Trans. Graph.*, 26(3), July 2007.

[NMK+06] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Comput. Graph. Forum*, 25(4):809–836, 2006.

[NSACO05] A. Nealen, O. Sorkine, M. Alexa, and D. Cohen-Or. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph.*, 24(3):1142–1147, 2005.

[NVW+13] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Trans. Graph.*, 32(6):179:1–179:10, 2013.

[OBP+12] V. Orvalho, P. Bastos, F. Parke, B. Oliveira, and Alvarez X. A facial rigging survey. In *in Proc. of the 33rd Annual Conference of the European Association for Computer Graphics - Eurographics*, pages 10–32, 2012.

[OBP+13] C. A. Öztireli, I. Baran, T. Popa, B. Dalstein, R. W. Sumner, and M. Gross. Differential blending for expressive sketch-based posing. In *Proceedings of the 2013 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 155–164, 2013.

[OMA09] V.C. Orvalho, J. Miranda, and Sousa A.A. What a feeling: Learning facial expressions and emotions. In *Videojogos Conference*, November 2009.

[O'N08] R. O'Neill. *Digital Character Development: Theory and Practice.* Morgan Kaufmann, Elsevier, 2008.

[Osi07] J. Osipa. *Stop Staring: Facial Modeling and Animation Done Right.* Wiley Publishing, Sybex, 2 edition, 2007.

[OSS07] L. Olsen, F. F. Samavati, and M. C. Sousa. Fast stroke matching by angle quantization. In *Proceedings of the First International Conference on Immersive Telecommunications*, ImmersCom '07, pages 6:1–6:6,

ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[OSSJ05] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-Based Mesh Augmentation. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2005.

[OSSJ08] L. Olsen, F. F. Samavati, M. Costa Sousa, and J. Jorge. A Taxonomy of Modeling Techniques using Sketch-Based Interfaces. In *Eurographics 2008 - State of the Art Reports*. The Eurographics Association, 2008.

[Par72] F. I. Parke. Computer generated animation of faces. In *Proc. ACM National Conference*, 1972.

[Par74] F. I. Parke. *A Parametric Model for Human Faces*. Ph.D. Thesis, University of Utah, Salt Lake City, 1974.

[PF05] Lewis J.P. Pighin F. Digital face cloning. SIGGRAPH Course, 2005.

[PF06] Lewis J.P. Pighin F. Performance-driven facial animation. SIGGRAPH Course, 2006.

[PHL+98] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 75–84, New York, NY, USA, 1998. ACM.

[Pie87] L. Piegl. Interactive data interpolation by rational bezier curves. *IEEE Comput. Graph. Appl.*, 7(4):45–58, April 1987.

[PJBF00] J. P. Pereira, J. A. Jorge, V. A. Branco, and F. N. Ferreira. Towards calligraphic interfaces: Sketching 3d scenes with gestures and context icons. In *WSCG*, 2000.

[Pro96] X Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *In Graphics Interface*, pages 147–154, 1996.

[PSNW07] R. Pusch, F. Samavati, A. Nasri, and B. Wyvill. Improving the sketch-based interface: Forming curves from many small strokes. *The Visual Computer*, 23(9-11):955–962, Sep 2007.

[PSS04] H. Pyun, H. J. Shin, and S.Y. Shin. On extracting the wire curves from multiple face models for facial animation. *Computers and Graphics*, 28(5):757–765, 2004.

[Pug92] D. Pugh. Designing solid objects using interactive sketch interpretation. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, I3D '92, pages 117–126, New York, NY, USA, 1992. ACM.

[Rai04] B. Raitt. The making of gollum. Presentation at University of Southern California Institute for Creative Technologies Frontiers of Facial Animation Workshop, 2004.

[RCB05] K. Ritchie, J. Callery, and K. Biri. *The Art of Rigging: A Definitive Guide to Character Technical Direction with Alias Maya, Volume 1*. CG ToolKit, Alias, 2005.

[RF16] N. A. Rumman and M. Fratarcangeli. Skinning techniques for articulated deformable characters. *Computer Graphics Theory and Applications*, 2016.

[RHKK11] T. Rhee, Y. Hwang, J. D. Kim, and C. Kim. Real-time facial animation from live video tracking. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 215–224, New York, NY, USA, 2011. ACM.

[RSW+07] K. Rose, A. Sheffer, J. Wither, M.P. Cani, and B. Thibert. Developable surfaces from arbitrary sketched boundaries. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 163–172, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[SC04] A. Shesh and B. Chen. SMARTPAPER: An Interactive and User Friendly Sketching System. *Computer Graphics Forum*, 2004.

[SD07] T. M. Sezgin and R. Davis. Scale-space based feature point detection for digital ink. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[SF98a] K. Singh and E. Fiume. Wires: A geometric deformation technique. In *Proc. of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998.

[SF98b] K. Singh and E. Fiume. Wires: A geometric deformation technique. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 405–414, 1998.

[SGsGU02] E. Saykol, G. Güle¸sir, U. Gudukbay, and Ö. Ulusoy. Kimpa: A kinematics-based method for polygon approximation. In *Advances in Information System (ADVIS'2002)*, volume 2457 of *Lecture Notes in Computer Science*, pages 186–194. Springer, October 2002.

[SGT08] R. Schmedding, M. Gissler, and M. Teschner. Optimized damping for dynamic simulations. In *Proceedings of the 25th Spring Conference on Computer Graphics*, pages 189–196, 2008.

[SI07] H.J. Shin and T. Igarashi. Magic canvas: Interactive design of a 3-d scene prototype from freehand sketches. In *Proceedings of Graphics Interface 2007*, GI '07, pages 63–70, New York, NY, USA, 2007. ACM.

[SILN11] J. Seo, G. Irving, J. P. Lewis, and J. Noh. Compression and direct manipulation of complex blendshape models. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 164:1–164:10, 2011.

[SK00] K. Singh and E. Kokkevis. Skinning characters using surface-oriented free-form deformation. In *Graphics Interface*, pages 35–42. Citeseer, 2000.

[SKP08] S. Sueda, A. Kaufman, and D. K. Pai. Musculotendon simulation for hand animation. *ACM Trans. Graph.*, 27(3):83:1–83:8, August 2008.

[SMA99] F.F. Samavati and N. Mahdavi-Amiri. A filtered b-spline model of scanned digital images. *Journal of Science*, 10(4):258–264, 1999.

[SMND08] T. Sucontphunt, Z. Mo, U. Neumann, and Z. Deng. Interactive 3d facial expression posing through 2d portrait manipulation. In *Proceedings of Graphics Interface 2008*, 2008.

[SNF05] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. In *Proc. SIGGRAPH Papers*, 2005.

[SOG08] D. Steinemann, M. A. Otaduy, and M. Gross. Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2008.

[Sor06] O. Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, 2006.

[SP86] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proc. Computer Graphics and Interactive Techniques*, 1986.

[SSB13] F. S. Sin, D. Schroeder, and J. Barbic. Vega: Non-Linear FEM Deformable Object Simulator. *Computer Graphics Forum*, 2013.

[SSBT08] T. Stumpp, J. Spillmann, M. Becker, and M. Teschner. A geometric deformation model for stable cloth simulation. In *Workshop on Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2008)*, 2008.

[SSD06] T. M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: Early processing for sketch understanding. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[SSMCP02] J. Schleifer, R. Scaduto-Mendola, Y. Canetti, and M. Piretti. Character setup from rig mechanics to skin deformations: A practical approach. In *Proc. SIGGRAPH '02 Course Notes*, 2002.

[SSP07] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3), July 2007.

[SSRMF06] E. Sifakis, A. Selle, A. Robinson-Mosher, and R. Fedkiw. Simulating speech with a physics-based facial muscle model. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006.

[SSS06] A. Severn, F. Samavati, and M. C. Sousa. Transformation Strokes. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. The Eurographics Association, 2006.

[Sta09] J. Stam. Nucleus: Towards a unified dynamics solver for computer graphics. In *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics '09. 11th IEEE International Conference on*, pages 1–11, 2009.

[Sut64] I. E. Sutherland. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop*, DAC '64, pages 6.329–6.346, New York, NY, USA, 1964. ACM.

[SWSJ06] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[SZGP05] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović. Mesh-based inverse kinematics. *ACM Trans. Graph.*, 24(3):488–495, July 2005.

[TB96] D. Tolani and N.I. Badler. Real-time inverse kinematics of the human arm. *Presence*, 5(4):393–401, 1996.

[TBSR04] S. Tsang, R. Balakrishnan, K. Singh, and A. Ranjan. A suggestive interface for image guided 3d sketching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 591–598, New York, NY, USA, 2004. ACM.

[TBvdP04] M. Thorne, D. Burke, and M. van de Panne. Motion doodles: An interface for sketching character motion. *ACM Trans. Graph.*, 23(3):424–431, August 2004.

[TDlTM11] J. R. Tena, F. De la Torre, and I. Matthews. Interactive region-based linear 3d face models. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 76:1–76:10, 2011.

[THMG04] M. Teschner, B. Heidelberger, M. Müller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Proceedings of the Computer Graphics International*, 2004.

[TKZ+05] M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, . Raghupathi, A. Fuhrmann, M.P. Cani, F. Faure, N. Magnenat-Thalmann, and W. Strasser. Collision detection for deformable objects. *Computer Graphics Forum*, 24(1):61–81, 2005.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4):205–214, 1987.

[vFTS06] W. von Funck, H. Theisel, and H.P. Seidel. Vector field based shape deformations. *ACM Trans. Graph.*, 25(3):1118–1125, 2006.

[VMS05] P. A. C. Varley, R.R. Martin, and H. Suzuki. Frontal geometry from sketches of engineering objects: is line labelling necessary? *Computer-Aided Design*, 37(12):1285–1307, 2005.

[VTMS04] P. A. C. Varley, Y. Takahashi, J. Mitani, and H. Suzuki. A Two-Stage Approach for Interpreting Line Drawings of Curved Objects. In *Sketch Based Interfaces and Modeling*. The Eurographics Association, 2004.

[Wat87] K. Waters. A muscle model for animating three-dimensional facial expression. In *Proc. SIGGRAPH '87*, 1987.

[WAT+11] C. A. Wilson, O. Alexander, B. Tunwattanapong, P. Peers, A. Ghosh, J. Busch, A. Hartholt, and P. Debevec. Facial cartography: Interactive scan correspondence. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 205–214, New York, NY, USA, 2011. ACM.

[WB97] A. Witkin and D. Baraff. Physically based modeling: Principles and practice. In *ACM Siggraph 1997 Course notes*, 1997.

[WBLP11] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Real-time performance-based facial animation. In *Proc. SIGGRAPH Papers*, 2011.

[Wel93] C. Welman. Inverse kinematics and geometric constraints for articulated figure animation. Master Thesis, Simon Fraser University, 1993.

[WF95] K. Waters and J. Frisbee. A coordinated muscle model for speech animation. In *Graphics Interface '95*, pages 163–170, Ontario, Canada, May 1995. Canadian Human-Computer Communications Society.

[WFJ+05] B. Wyvill, K. Foster, P. Jepp, R. Schmidt, M. C. Sousa, and J. A. Jorge. Sketch based construction and rendering of implicit models. In *Proceedings of the First Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'05, pages 67–74, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[Wil90] L. Williams. Performance-driven facial animation. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, pages 235–242, New York, NY, USA, 1990. ACM.

[WOR10] H. Wang, J. O'Brien, and R. Ramamoorthi. Multi-resolution isotropic strain limiting. *ACM Trans. Graph.*, 29(6), 2010.

[WP02] X. C. Wang and C. Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proc. of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002.

[WPP07] R. Y. Wang, K. Pulli, and J. Popović. Real-time enveloping with rotational regression. *ACM Trans. Graph.*, 26(3), July 2007.

[WT92] K. Waters and D. Terzopoulos. The computer synthesis of expressive faces. *Philosophical Transactions: Biological Sciences*, pages 87–93, 1992.

[WXX+06] Y. Wang, Y. Xiong, K. Xu, K. Tan, and G. Guo. A mass-spring model for surface mesh deformation based on shape matching. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia*, pages 375–380, 2006.

[YL14] H. Yu and H. Liu. Regression-based facial expression optimization. *IEEE Transactions on Human-Machine Systems*, 44(3):386–394, June 2014.

[YSvdP05] C. Yang, D. Sharon, and M. van de Panne. Sketch-based modeling of parameterized objects. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[YZ06] X. Yang and J. Zhang. Stretch it - realistic smooth skinning. In *Proc. of the International Conference on Computer Graphics, Imaging and Visualisation, IEEE Computer Society*, 2006.

[ZB94] J. Zhao and N. I. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Trans. Graph.*, 13(4):313–336, October 1994.

[ZBO13] J. S. Zurdo, J. P. Brito, and M. A. Otaduy. Animating wrinkles by example on non-skinned cloth. *IEEE Trans. Vis. Comput. Graph.*, 19(1):149–158, 2013.

[ZHH06] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: An interface for sketching 3d scenes. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[ZLG+06] Q. Zhang, Z. Liu, B. Guo, D. Terzopoulos, and H.Y. Shum. Geometry-driven photorealistic facial expression synthesis. *IEEE Trans. Vis. Comput. Graph.*, 12(1):48–60, 2006.

[ZNA07] J. Zimmermann, A. Nealen, and M. Alexa. Silsketch: Automated sketch-based editing of surface meshes. In *Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling*, 2007.

[ZSCS04] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High resolution capture for modeling and animation. *ACM Trans. Graph.*, 23(3):548–558, August 2004.

[ZSTB10] Y. Zhu, E. Sifakis, J. Teran, and A. Brandt. An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Trans. Graph.*, 29(2):16:1–16:18, April 2010.